

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials

**SISTEMA DE CONTROL
D'UNA BUTACA MOTORITZADA
DES D'UN SMARTPHONE ANDROID**

MEMÒRIA

Autor: Sílvia Jané March
Director: Emili Lupon Rosés
Convocatòria: Gener 2017



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

Aquest projecte consisteix en el desenvolupament d'una aplicació de Blynk per a smartphone Android que permeti el control de les posicions del reposapeus i del respatllet d'una butaca motoritzada, a través d'Internet. La realització d'aquest projecte s'emmarca en l'àmbit d'Internet of Things, Open Source i Open Hardware.

Els objectius que es proposen són que l'smartphone permeti ajustar i monitoritzar les posicions del reposapeus i del respatllet i, a més a més, que l'usuari tingui la disponibilitat de guardar les posicions desitjades d'aquestes dues parts mòbils de la butaca. Per assolir aquests objectius es requereix programar una placa microcontroladora.

Es consideren els diferents blocs que agrupa tot el sistema proposat: un smartphone Android, una aplicació de control (Blynk), un servidor local de Blynk, un punt d'accés de WiFi, una placa Arduino UNO R3 i una butaca motoritzada.

L'aplicació MA_BUTACA dissenyada compleix tots els objectius proposats i les diverses opcions addicionals afegides. Es dóna èmfasi al nom de l'aplicació pel fet de treballar amb "*Mobile*" i "*Arduino*", i ha sigut tot un desafiament aconseguir la connexió entre els dos dispositius, la seva configuració i el seu paper per poder controlar els motors que determinen les posicions del reposapeus i del respatllet de la butaca.

Com a part final del projecte, s'ha desenvolupat un prototipus demostrador per poder visualitzar les diferents opcions de control que ofereix MA_BUTACA.





Sumari

RESUM	1
SUMARI	3
GLOSSARI	7
PREFACI	11
Origen del projecte	11
Motivació.....	11
Requeriments previs.....	11
1. INTRODUCCIÓ	13
1.1. Objectius del projecte.....	14
1.2. Abast del projecte	14
2. DISSENY DEL SISTEMA	17
2.1. Proposta del sistema.....	17
2.2. Apps de control per a smartphone.....	18
2.2.1. MIT App Inventor	18
2.2.2. Blynk	18
2.2.3. Selecció de l'App de control.....	19
2.3. Hardware de control.....	19
2.3.1. Arduino	20
2.3.2. Raspberry Pi	21
2.3.3. Comparació i selecció del hardware de control	22
2.4. Motors per a les butaques motoritzades.....	23
2.4.1. Motors DC.....	24
2.4.2. Motor pas a pas	24
2.4.3. Servomotor	25
2.4.4. Selecció dels motors.....	26
3. COMPONENTS DEL SISTEMA	27
3.1. Descripció del sistema desenvolupat.....	27
3.2. Arduino UNO R3	28
3.3. Connectivitat d'Arduino a Internet.....	30
3.3.1. Arduino UNO WIFI.....	30
3.3.2. Mòdul WiFi (ESP8266)	31
3.3.3. Cable USB connectat al PC.....	34



3.4.	Servomotors SunFounder	35
3.4.1.	Font d'alimentació YwRobot	36
3.5.	Sensor de presència.....	37
3.6.	Descripció del sistema conjunt.....	38
4.	EINES DE PROGRAMACIÓ I LLIBRERIES	39
4.1.	Eines de Blynk	39
4.2.	Arduino IDE	41
4.3.	Metodologia per crear un projecte amb Blynk per Arduino	44
4.4.	Exemple de prova de funcionament de Blynk.....	46
5.	DISSENY I DESENVOLUPAMENT DEL SISTEMA	49
5.1.	Disseny i desenvolupament de l'aplicació.....	49
5.2.	Descripció de l'aplicació MA_BUTACA a l'entorn BLYNK	56
5.2.1.	Aplicació MA_BUTACA per al desenvolupament	57
5.2.2.	Aplicació MA_BUTACA per l'usuari	58
5.3.	Codi de funcionament de MA_BUTACA a l'Arduino	59
5.4.	Exemple de funcionament del sistema	61
6.	MANUAL D'UTILITZACIÓ DE MA_BUTACA	67
7.	PRESSUPOST	71
7.1.	Costos Directes	71
7.1.1.	Costos de recursos materials	71
7.1.2.	Costos de recursos humans.....	72
7.2.	Costos indirectes	72
7.3.	Cost total del projecte	73
8.	IMPACTE AMBIENTAL	75
9.	CONCLUSIONS	77
10.	EXTENSIONS FUTURES	79
10.1.	Servidor local: Blynk server	79
10.2.	Mòdul ESP8266.....	81
10.3.	Identificador de butaca	81
10.4.	Usabilitat per a tot tipus d'usuaris.....	82
AGRAÏMENTS		83
BIBLIOGRAFIA.....		85
Referències bibliogràfiques.....		85





Glossari

APL: *A Programming Language*, llenguatge de programació.

DIY: *Do It Yourself*, és la filosofia que rebutja tenir que comprar tot i substituir el treball que es pot realitzar pels nostres propis mitjans.

EEPROM: *Electrically-Erasable Programmable Read-Only Memory*, és un tipus de memòria permanent que poden esborrar mitjançant procediments elèctrics.

GND: *Ground*, és el punt de referència d'un circuit elèctric.

GPIO: *General Purpose Input Output*, és un pin genèric d'entrada i sortida, en un circuit integrat que es pot controlar per l'usuari en temps real.

HDMI: *High Definition Multimedia Interface*, és una norma per a la interconnexió d'àudio i vídeo digital sense comprimir.

HDTV: *High Definition Television*, és dels formats que es caracteritzen per emetre senyals amb una qualitat digital superior als sistemes tradicionals analògics de televisió en color.

ICSP: *In Circuit Serial Programming*, és l'habilitat d'alguns dispositius lògics programables, microcontroladors i altres circuits electrònics, de ser programats mentrestant estan instal·lats a un sistema complet, en lloc de requerir que el xip sigui programat abans de ser instal·lat dins del sistema.

IDE: *Integrated Development Environment*, és la interfície de desenvolupament per facilitar la programació i configuració de les aplicacions.

IDLE: *Integrated DeveLopment Environment*, és un entorn integrat de desenvolupament per al llenguatge Python.

IoT: *Internet of Things*, és un concepte que es refereix a la interconnexió digital dels objectes quotidians amb Internet.

IP adress: *Internet Protocol adress*, és el número que identifica cada dispositiu dins d'una xarxa d'ordinador.

JDK: *Java Development Kit*, és software que conté un conjunt d'eines de desenvolupament per la creació de programes en Java.



LED: *Light Emitting Diode*, díode emissor de llum.

MIT: *Massachusetts Institute of Technology*, és una institució universitària situada a la ciutat de Cambridge (Massachusetts, USA), que es dedica a la recerca i investigació.

NTSC: *National Television Committee*, és un sistema de transmissió del senyal de televisió.

PAL: *Phase Alternating Line*, és un sistema de transmissió de senyals analògics de televisió en color amb l'objectiu de millorar la qualitat i reduir els errors de fase que es produeixen en el sistema NTSC.

PC: *Personal Computer*.

PWM: *Pulse-Width Modulation*, és un senyal digital quadrat del qual es pot variar el cicle de treball sense variar la freqüència.

QEMU: *Quick Emulator*, és un emulador de terminal de processadors per programari lliure, que permet a un usuari simular un sistema complet dins d'un altre.

RAM: *Random Access Memory*, memòria d'accés aleatori d'escriptura i lectura. Es fa servir com a memòria de treball dels computadors.

RX: *Receiver*, receptor.

SD: *Secure Digital*, és un dispositiu en format de targeta de memòria per a dispositius portàtils.

SoC: *System on Chip*, és un tipus de dispositius que integra en un sol xip els diferents components del sistema.

TTL: *Transistor-Transistor Logic*, és una tecnologia de construcció de circuits electrònics digitals.

TX, *Transmission*, transmissió.

UART: *Universal Asynchronous Receiver-Transmitter*, és un xip de tipus receptor/transmissor asíncron que tradueix les dades entre format paral·lel i sèrie.

USB: *Universal Serial Bus*, és un bus estàndard industrial que defineix el cablejat, els connectors i el protocol de comunicacions emprat en un bus de dades per a connectar, comunicar i alimentar perifèrics des dels ordinadors.



WiFi: *Wireless Fidelity*, és una tecnologia de xarxa local sense fils que permet a un dispositiu electrònic intercanviar dades o connectar amb Internet.





Prefaci

Origen del projecte

Actualment estic realitzant pràctiques curriculars d'empresa i, gràcies a la meua activitat al departament d'Oficina Tècnica de l'empresa on les realitzo, tracto diversos temes d'enginyeria. Un d'ells ha estat l'electrònica dels diversos components de les butaques motoritzades i el seu control, que es fa actualment amb connexió per cable.

A partir d'aquí, va sorgir la idea de controlar una butaca des d'un telèfon mòbil, amb una connexió sense fils. Gràcies a aquesta idea, vaig proposar el tema d'aquest treball final de grau, amb la intenció d'estudiar la viabilitat d'un sistema de control d'una butaca motoritzada des d'un smartphone Android, a través d'Internet. El projecte s'ha realitzat dins dels meus coneixements i del temps limitat d'un treball final de grau.

Motivació

Buscava un tema relacionat amb l'electrònica o amb l'automàtica i que tingués afinitat amb la continuïtat dels meus estudis al màster d'Enginyeria Industrial. El tema d'aquest treball és un tema que em motiva, degut a què engloba tots els aspectes d'*Internet of Things* (IoT), d'Open Source i d'Open Hardware i és una bona oportunitat per familiaritzar-me en aquest àmbit. En un futur es podria valorar el desenvolupament d'aplicacions més complexes enfocant-les cap a la seva fabricació a nivell industrial.

També em motiva molt realitzar aquest projecte, ja que inclou el desenvolupament d'un prototipus demostrador amb components reals (mòbil Android, placa Arduino i servomotors), que permeti verificar el funcionament del sistema dissenyat.

Requeriments previs

Per poder realitzar aquest projecte es requereix tenir una sèrie de coneixements previs d'electrònica i de programació, degut a què aquest projecte engloba ambdós aspectes.

Es tenia coneixement previ del llenguatge Python, que és relativament semblant al llenguatge C/C++ amb el que s'ha desenvolupat la programació d'aquest projecte. Els coneixements d'electrònica es van adquirir durant el quadrimestre anterior a l'assignatura "Electrònica".





1. Introducció

Es vol crear una aplicació d'un smartphone que permeti governar el sistema de control d'una butaca motoritzada, a través d'Internet. La idea es basa principalment en crear un software perquè l'usuari pugui ajustar les posicions desitjades del reposapeus i respatllet.

Tot i que es podrien afegir moltes opcions, s'ha decidit enfocar el projecte a l'estudi de les funcionalitats del sistema de control d'una butaca motoritzada. Potser en un futur es podria acabar de completar tot l'estudi, afegint altres necessitats de l'usuari. El disseny d'aquest projecte es farà per a mòbils Android.

S'estudiaran les necessitats del sistema de control per a una butaca real. Però també es realitzarà un prototipus demostrador que utilitzarà els següents components principals: mòbil, desenvolupador d'Apps per smartphone, placa microcontroladora, protoboard, font d'alimentació i motors de posició. Per a la realització d'aquest projecte, es considerarà un hardware controlador orientat a projectes de tipus *Do It Yourself* (DIY).

Es desenvoluparà una aplicació de control dels motors per a un smartphone que:

- Controli la posició i velocitat dels motors que permeten ajustar les posicions del reposapeus i del respatllet.
- Monitoritzi les posicions del reposapeus i del respatllet.
- Tingui l'opció de guardar les posicions del reposapeus i del respatllet desitjades per l'usuari.

Amb aquestes prestacions es pretén que l'usuari tingui la màxima facilitat per utilitzar aquesta aplicació de mòbil i la màxima comoditat un cop que estigui assegut a la butaca. Es crearà una aplicació de mòbil intuïtiva i fàcil d'utilitzar, fins i tot per usuaris poc familiaritzats amb els mòbils.

Cal destacar la modernitat dels aspectes amb els que es treballa al llarg d'aquest projecte. És a dir, es fa servir l'àmbit d'IoT, del que es va parlar per primera vegada a l'any 1999, si bé fins l'any 2005 no va adquirir un nivell important d'utilitat i d'ús. En el mateix any, es van popularitzar les plaques hardware de control enfocades als projectes de DIY. El fet de que tots aquests àmbits siguin tant recents ajuda a recopilar la informació actual i adequada per a la realització del desenvolupament de l'aplicació.



1.1. Objectius del projecte

Es proposa una sèrie d'objectius per a aquest treball amb la finalitat d'aconseguir controlar una butaca motoritzada.

Els objectius generals d'aquest projecte són:

- Dissenyar i desenvolupar una aplicació per smartphone amb connexió a Internet, que permeti controlar les posicions del reposapeus i respatllet de la butaca i monitoritzar-les.
- Programar una placa microcontroladora que permeti seleccionar la posició del respatllet i reposapeus i controlar la seva velocitat de moviment.
- Crear un prototipus demostrador amb components reals (mòbil Android, placa Arduino i servomotors), que verifiqui el funcionament del sistema dissenyat.

Per desenvolupar aquests objectius generals, es proposen uns objectius específics i condicions relacionades amb el control de la butaca motoritzada:

- Moure tots els servomotors a la posició angular desitjada.
- Tenir la disponibilitat de guardar la posició desitjada dels elements de la butaca.
- Tenir l'opció de moure la butaca a la posició guardada.
- Posar les posicions màximes directament, i de manera ràpida, respectant el confort de l'usuari.
- Si no hi ha ningú assegut, la qual cosa es detectarà amb un sensor apropiat, la butaca ha de tornar a la posició inicial, fent un RESET.

1.2. Abast del projecte

En aquest projecte s'utilitzen coneixements d'electrònica i de programació per assolir els objectius proposats.

Es crea un prototipus demostrador amb components reals (mòbil Android, placa controladora i servomotors) per verificar el funcionament del sistema dissenyat. Aquest prototipus demostrador està preparat per aplicar-ho a una maqueta de butaca.



Cal recordar que aquest prototipus està orientat a demostrar la viabilitat del seu funcionament i no està preparat per a aplicacions industrials. El temps limitat que es disposa per desenvolupar aquest projecte no ho permet. No obstant, les proves realitzades i les conclusions obtingudes faciliten definir unes extensions futures per fer una possible aplicació industrial.





2. Disseny del sistema

2.1. Proposta del sistema

En aquest capítol es comenten els diferents blocs del sistema que es consideren al projecte. El sistema constarà de: a) un mòbil Android, b) una app de control, c) un servidor (núvol o local), d) un punt d'accés de WiFi, e) un controlador i f) una butaca motoritzada (Fig. 2.1). Es realitza un estudi dels requisits tècnics de cada bloc, i es fa una comparativa dels diversos components disponibles al mercat, amb l'objectiu d'escollir la millor opció tecnològica per a cadascun dels blocs del prototipus demostrador.

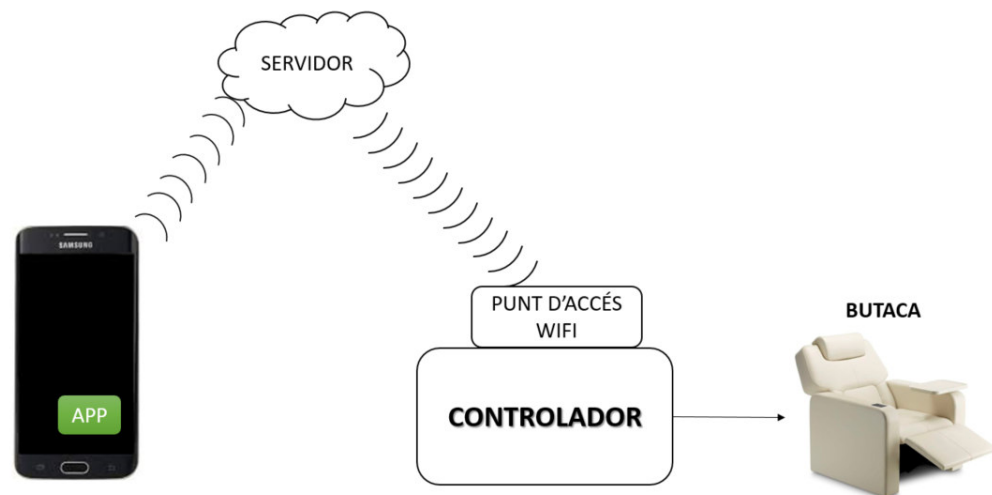


Figura 2.1. Blocs del sistema

En primer lloc, s'ha fet una recerca de diferents tipus d'aplicacions que permetin crear i desenvolupar una "App" dissenyada per poder controlar la butaca motoritzada.

Es proposa que el mòbil estigui connectat per via WiFi o dades, amb un servidor d'Internet (núvol o local). Per altra banda, es considera que el controlador es pugui connectar al servidor a través d'un punt d'accés WiFi. S'estudiaran les diverses opcions per poder aconseguir aquest tipus de connexió i comunicació.

El següent pas ha estat buscar els diferents tipus de controladors compatibles amb l'aplicació escollida. Les principals característiques d'aquest bloc són la facilitat per al control dels motors, la possibilitat de connexió WiFi, la disponibilitat de llibreries per desenvolupar la seva programació i la comunicació amb el mòbil. S'ha observat que al mercat hi ha models de controladors molt diferents, respecte al xip microcontrolador o la



capacitat de connectar sensors o actuadors. Per tant, es farà una cerca dels diversos tipus de controladors comparant les seves especificacions tècniques.

Finalment, l'últim pas ha sigut la cerca d'informació sobre les característiques de les butaques motoritzades i els motors que utilitzen, per tal que es pugui moure el respall i el reposapeus una determinada longitud o un determinat angle. També s'ha estudiat el tipus de motor que es podria utilitzar al prototipus demostrador, per verificar el funcionament del sistema.

2.2. Apps de control per a smartphone

S'utilitza un smartphone Android per poder controlar les posicions del reposapeus i del respall. S'ha fet una recerca dels diferents tipus d'aplicacions per crear una App en aquest sistema operatiu.

Entre els tipus de desenvolupadors, els més destacats són el MIT App Inventor i el Blynk. Fins i tot, hi ha una tercera possibilitat que permet desenvolupar l'App a través del Java. No obstant, aquesta opció s'ha descartat degut a què el seu procediment és molt complex i requereix disposar de coneixements profunds de Java.

Per tant, es procedeix a fer una comparativa entre aquests dos tipus d'aplicacions.

2.2.1. MIT App Inventor

L'aplicació MIT App Inventor [5] és un entorn de desenvolupament de software creat per Google Labs i el *Massachusetts Institute of Technology* (MIT), per a la creació d'aplicacions mòbils per al sistema operatiu Android. L'usuari pot enllaçar una sèrie de blocs per crear l'aplicació de forma visual i a partir d'un conjunt d'eines bàsiques. Està format per tres eines: gestor de projectes, dissenyador i editor de blocs. L'App Inventor està dirigida a les persones que no estan familiaritzades amb la programació informàtica. La connexió entre el mòbil i l'ordinador per crear l'aplicació es fa a través de connexió WiFi. Però, per fer una aplicació pràctica de control es necessita un mòdul addicional, Bluetooth, per establir la connexió entre el mòbil i una placa hardware controladora degut a què no hi ha els drivers disponibles per a WiFi.

2.2.2. Blynk

Blynk [1] és un desenvolupador d'Apps, que va ser creat per Pasha Baiborodin [10], expert en enginyeria de software. La seva inspiració va ser realitzar una aplicació de mòbil que permetés controlar la placa Arduino des d'un smartphone i després es va generalitzar a altres plataformes. Es va poder desenvolupar gràcies el suport que va rebre des de



Kickstarter [9], que és una organització de Crowdfunding on es recol·lecta diners per donar suport econòmic als nous projectes creatius. Finalment, aquesta aplicació Blynk es va llançar públicament al maig de 2015 [3].

Blynk és una plataforma compatible amb els sistemes operatius iOS i Android, que permet controlar plaques com Arduino o Raspberry Pi a través d'Internet. Blynk ofereix un tauler d'instruments digitals on es pot construir una interfície gràfica simplement arrossegant i deixant anar els widgets i assignant a la placa uns pins analògics, digitals i/o virtuals. L'aplicació Blynk suporta diferents tipus de plaques, com Arduino, Raspberry, ESP8266, etc. i es connecta a Internet a través de WiFi, Ethernet o dades telefòniques. Blynk permet aconseguir la connexió amb aquestes plaques, a través d'un servidor (núvol o local), amb una direcció predefinida als dos dispositius, mòbil i controlador. Blynk està preparat per a tot tipus d'aplicacions d'IoT.

2.2.3. Selecció de l'App de control

Ambdues aplicacions són gratuïtes i es poden descarregar fàcilment des de la web. També són fàcils de manipular.

La principal raó per la qual es decideix emprar Blynk, és que MIT App Inventor no disposa de drivers per a WiFi. A més a més, Blynk està preparada per controlar diverses plaques de hardware, de manera més fàcil, ràpida i directa. També, quan es vol d'introduir i configurar els widgets al tauler d'instruments digitals només es necessita indicar a quin pin, real o virtual, del hardware es troba connectat. Com a factor positiu, cada widget ofereix només la connexió als tipus de pins limitats per aquella funció. Així doncs, s'evita realitzar grans errors. En el cas del MIT App Inventor, es parteix des de zero en la creació del projecte i requereix estructurar l'App a través d'un esquema de blocs. Per tant, per al que es necessita en aquest projecte no era una opció viable i profitosa, ja que el seu disseny és més complex i complicat per crear l'App.

A més a més, la plataforma Blynk està preparada per desenvolupar les aplicacions amb uns widgets de manera molt intuïtiva i directa, que facilita el seu ús per tot tipus d'usuaris.

2.3. Hardware de control

Per controlar la butaca motoritzada s'ha optat per fer una recerca exhaustiva de diferents tipus de hardware que siguin compatibles amb l'aplicació Blynk que s'ha escollit, tal com s'ha comentat en l'apartat anterior.



Es requereix una placa hardware per poder controlar la butaca motoritzada de dos motors, el respall i el reposapeus. Per això, s'ha buscat els diferents tipus de hardware més coneguts i utilitzats al mercat, que siguin compatibles amb el desenvolupador Blynk, dels quals els més destacats són Arduino i Raspberry Pi. Es farà una comparativa de les especificacions tècniques d'ambdós dispositius per extreure conclusions i decidir quina placa encaixa millor per aquest treball.

2.3.1. Arduino

L'Arduino [2] és una plataforma electrònica de codi obert (open-source) composta de hardware i software per a dissenyar, desenvolupar i realitzar proves als prototipus d'electrònica i productes complexos. El hardware consisteix en un microcontrolador amb altres components electrònics que poden ser programats, utilitzant el software, per fer qualsevol tasca. La programació d'Arduino està basada en el llenguatge C/C++, que permet que el seu ús sigui fàcil per a qualsevol usuari aficionat a l'electrònica. Cal destacar que hi ha una gran varietat de llibreries desenvolupades per moltes aplicacions.



Figura 2.2. Placa Arduino UNO R3

La naturalesa de l'open-source permet que tot el hardware, i les llibreries relacionades amb les seves aplicacions, estiguin disponibles per l'ús personal o comercial. A més a més, no es requereix cap tipus de llicència per al seu desenvolupament, ús, redistribució o fins i tot la venda, a excepció del nom ArduinoTM, que és una marca registrada.

El hardware d'aquesta plataforma conté un circuit integrat incorporat: el microcontrolador de tipus AVR, que necessita ser programat per realitzar la tasca desitjada. Els programes que es desenvolupen utilitzen l'APL (*A Programming Language*). Aquests, després de compilar-se, es carreguen en la memòria del microcontrolador de l'Arduino. Per a qui no sigui un professional, escriure programes no és una tasca fàcil. Per això l'objectiu principal de l'Arduino és disminuir la dificultat en la programació dels algorismes.



El model bàsic de l'Arduino disposa d'un microcontrolador ATmega328P de 8 bits, un regulador de 5 V, connectors de sortides, adaptador d'entrada, etc. Els ports d'entrada-sortida estan preparats de manera que la connexió amb els mòduls externs sigui fàcil i accessible. La plataforma permet interactuar amb sensors, circuits o altres perifèrics.

La plataforma Arduino disposa d'un entorn de desenvolupament integrat, IDE (*Integrated Development Enviroment*), que corre sobre un PC i que consisteix d'un editor per escriure els programes ("sketch") a executar en el microcontrolador de la placa hardware, d'un compilador i d'eines de comunicació amb la placa hardware.

Al compilar, el programa "sketch" és combina amb les funcions requerides de les llibreries estàndard de l'Arduino i les que han estat afegides per l'usuari. A continuació, el programa compilat és transferit al microcontrolador. Aquesta transferència es pot realitzar a través d'un cable USB o del port sèrie adequat del PC. L'Arduino conté un ICSP (*In-Circuit Serial Programming*) que permet als usuaris transferir el software dins del microcontrolador sense la necessitat d'extreure el circuit.

2.3.2. Raspberry Pi

El Raspberry Pi [13] és una placa que treballa com una computadora completament funcional. Utilitza llenguatges d'alt nivell com el Python, C++ i Java. Amb altres paraules, és una computadora en miniatura de baix cost i dissenyada inicialment per als estudiants, per a interactuar amb els seus computadors d'una manera molt fàcil.

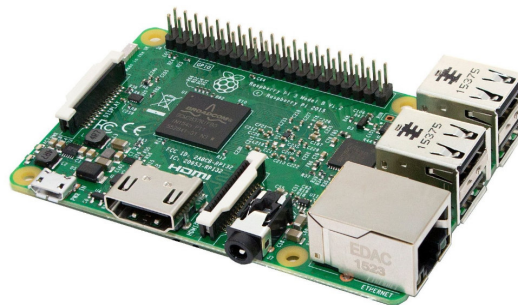


Figura 2.3. Placa Raspberry Pi B

Raspberry Pi utilitza un controlador Broadcom, què és un SoC (*System on Chip*) basat en un processador molt potent, l'ARM11, que té una velocitat de 700MHz. Aquesta minicomputadora no conté un display, però pot ser utilitzada amb un display HDTV i els estàndards de TV NTSC o PAL. Conté un port Ethernet que permet connectar-la a la xarxa.

Com a sistemes de desenvolupament d'aplicacions, es poden utilitzar ordinadors amb diferents sistemes operatius, com el Mac, Windows i Linux, sempre i quan es tingui el software adequat.



Raspberry Pi està enfocat, bàsicament, per a projectes multimèdia complexos o basats en Linux.

2.3.3. Comparació i selecció del hardware de control

Un cop s'han comentat les principals característiques de cada plataforma als dos apartats anteriors, es comparen les especificacions tècniques de l'Arduino UNO R3 i Raspberry Pi B. A continuació, es valorarà quina plataforma seria la més adequada per aquest projecte.

	ARDUINO UNO R3	RASPBERRY PI B
Processador	ATmega328P	Arm 11
Velocitat	16 MHz	700 MHz
RAM	2 KB	256 MB
USB	1	2
Flash	32 KB	(SD Card)
EEPROM	1 KB	-
Input Voltage	7 – 12 V	5 V
Audio	-	HDMI, Analògic
Video	-	HDMI, Analògic
Ethernet	-	10/100
I/O	14 GPIO, 6-10 bits analògics	8 GPIO
Sistema Operatiu	-	Linux
Entorn	Arduino IDE	Linux, IDLE, Open-Embedded, QEMU, Scratchbox, Eclipse
Tamany	7.6 x 1.9 x 6.4 cm	8.6 x 5.4 x 1.7 cm
Cost	23,70 €	40 €

Taula 2.1. Taula d'especificacions tècniques



Seguidament, comparant les especificacions tècniques i observant les principals funcions i aplicacions de les dues plataformes, s'ha pres la decisió d'utilitzar la placa Arduino. El principal avantatge de l'Arduino és que disposa d'un microcontrolador i múltiples entrades i sortides digitals, que el fa ser més apropiat per a la realització d'aquest projecte, mentre que el Raspberry Pi disposa d'un processador més potent, però més enfocat a projectes de multimèdia o basats en Linux.

2.4. Motors per a les butaques motoritzades

Les butaques individuals motoritzades cada cop tenen un paper més important al mercat i estan dirigides per a tot tipus d'usuari. Permeten obtenir una posició desitjada del respatllet i del reposapeus. El control es realitza normalment a través d'una botonera situada en el reposa braços, i connectada per cable, amb unes indicacions d'ús fàcil d'emprar, per regular la posició. En general, els motors de les butaques són actuadors lineals ja que disposen d'una potència elevada que permet elevar, ajustar, inclinar o empènyer als subjectes que estan asseguts.

S'observa que els diferents tipus de butaques motoritzades en el mercat tenen un angle de gir comprés aproximadament entre 0° fins a 18° - 55° en el respatllet, i fins a 55° - 108° en el reposapeus (Fig. 2.4)[4][11].

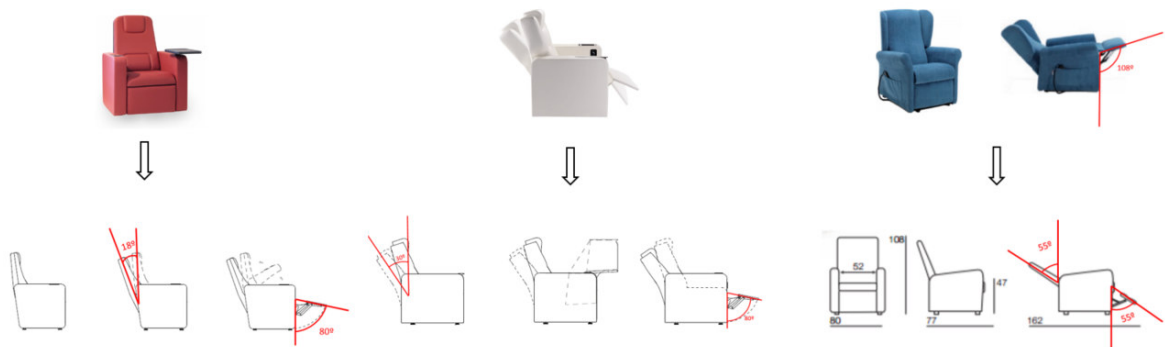


Figura 2.4. Exemples de butaques motoritzades

Es va començar a buscar diferents actuadors lineals, però eren només per a aplicacions industrials. Com que en aquest projecte es vol crear un prototipus demostrador s'han buscat diferents tipus de motors petits que permetin ajustar la posició desitjada per l'usuari. Els principals motors que es poden aplicar en un prototipus demostrador podrien ser els següents: motor DC, motor pas a pas i servomotor de posició [7].



Tenint en compte que s'ha escollit treballar amb la placa Arduino, es fa una cerca de les especificacions tècniques dels diferents tipus de motors compatibles amb aquesta placa.

2.4.1. Motors DC

Un motor de corrent continua, com altres motors elèctrics, converteix l'energia elèctrica en energia mecànica. Es compon de dues parts, l'estator i el rotor. Considerant un rotor amb debanats i un estator amb debanats o imants permanents, quan el corrent elèctric circula pel debanat del rotor, es crea un camp electromagnètic. Aquest interactua amb el camp magnètic creat per l'estator, provocant un rebuig entre els pols d'igual signe dels camps magnètics de l'estator i del rotor, creant un parell de força on el rotor gira en un sentit donat. Si es vol canviar el sentit de gir del rotor, s'ha de canviar el sentit del corrent que se li proporciona al rotor, només cal invertir la polaritat de la pila o bateria amb la que s'alimenta el debanat del rotor.

Per controlar un motor DC des d'Arduino, ens cal utilitzar un driver per a motors que proporcioni més corrent al motor, ja que els pins d'entrades i sortides de l'Arduino només generen 20 mA, corrent habitualment inferior a la requerida pels debanats d'un motor. D'aquesta manera, s'haurà d'alimentar el motor amb una font d'alimentació externa.

El driver L293D és un circuit integrat orientat al control de motors DC mitjançant ponts en H. Un pont en H és un sistema per controlar el sentit del gir d'un motor DC utilitzant quatre transistors. Els transistors es comporten com a interruptors i, depenent de quins transistors condueixen i quins no, canvia la polarització, és a dir, el sentit de gir. Aquest driver conté dos ponts en H, proporciona un corrent de 600 mA al motor i suporta un voltatge entre 4,5 V i 36 V.

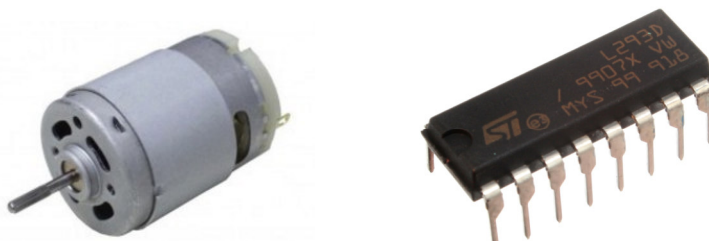


Figura 2.5. Motor DC i driver L293D

2.4.2. Motor pas a pas

Un motor pas a pas (també anomenat “stepper”) és un dispositiu electromagnètic que converteix els impulsos elèctrics en moviments mecànics de rotació. La principal característica d'aquests motors és que es mouen un cert angle, anomenat pas, per cada impuls de corrent que reben. Normalment els passos poden ser de 1,8° a 90° per pas, depenent del motor. Són motors amb molta precisió, que permeten quedar fixos en una



posició, com els servomotors, i també són capaços de girar lliurement en un sentit o un altre, com els motors DC. Els motors pas a pas estan formats per dues parts, l'estator i el rotor, que van muntades sobre un eix.

Hi ha tres tipus de motors pas a pas: imant permanent, reluctància variable i híbrid.

Per controlar aquest tipus de motor des de l'Arduino també es necessita un driver L293D.



Figura 2.6. Motor pas a pas i driver L293D

2.4.3. Servomotor

Un servomotor, també anomenat servo, és un sistema complex basat en un motor de contínua que permet assolir una posició indicada dins del seu rang d'operació i mantenir-se fix en ella. Normalment l'angle es pot variar de 0 a 180 graus i s'alimenta amb una tensió de 5 V mínim.

El servomotor està format per un motor de corrent contínua, una caixa reductora, un joc d'engrenatges, un potenciòmetre i un circuit de control, on té integrat un pont H (Fig. 2.8). Pot aguantar una quantitat determinada de pes a través del parell del servomotor. Per controlar un servomotor, s'utilitzen tècniques de PWM (Modulació per amplada de polsos). La majoria treballa a una freqüència de 50 Hz (període de 20 ms). Quan s'envia un pols, l'amplada d'aquest determina la posició angular del servomotor. L'amplada varia segons el servomotor, però acostuma a estar compresa entre 0,5 ms i 2,5 ms.



Figura 2.7. Servomotor

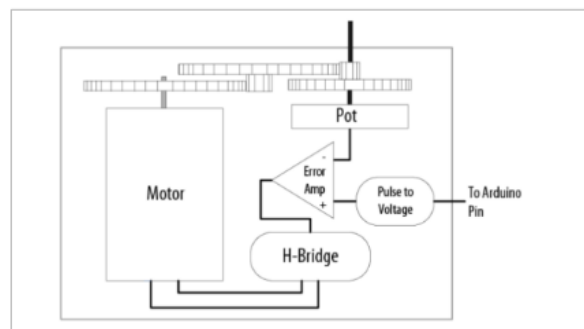


Figura 2.8 Esquema de blocs del servo



2.4.4. Selecció dels motors

Les butaques motoritzades comercials utilitzen uns actuadors lineals que permeten moure les posicions del respall i reposapeus. Disposen de molta potència per poder desplaçar la butaca a aquestes posicions amb l'usuari assegut.

De totes maneres, per a la realització del prototipus demostrador, s'han considerat diverses alternatives per escollir quin és el motor més apropiat.

S'observa que el motor DC no és un bon candidat perquè la seva funció principal és girar i no precisar cap angle de posició. En aquest cas, els motors pas a pas i el servomotors són els que encaixen millor. L'objectiu principal és crear un prototipus amb els mínims components electrònics possibles i el mínim cost.

Per tant, s'escull els servomotors pel fet que tenen tots els components integrats dins de la caixa del servo i es poden connectar directament a l'Arduino. Tot i que en aquest cas, s'alimentaran d'una font externa addicional per tal d'evitar errors d'arrencada, ja que en aquest projecte s'utilitzaran dos servomotors.

Es poden trobar servomotors amb un cost de 12 €. Mentre que els motors pas a pas tenen preus al voltant de 14€ i requereixen un driver com el L293D, que val 10 €. Aquest és un preu aproximat, perquè depenent d'on es compra, es poden trobar millors ofertes.



3. Components del sistema

En aquest capítol, s'explica breument quins són els components escollits, les seves característiques rellevants, i la configuració general del sistema. Tota la descripció està orientada al disseny i realització del prototipus demostrador proposat.

3.1. Descripció del sistema desenvolupat

El sistema dissenyat en aquest projecte està format per (Fig. 3.1):

- Smartphone que disposa de connexió de dades o WiFi.
- Arduino UNO R3.
- Dispositius d'accés a Internet (mòdul ESP8266 o PC).
- Servomotors.
- Font d'alimentació.
- Sensor de presència (interruptor i LED).

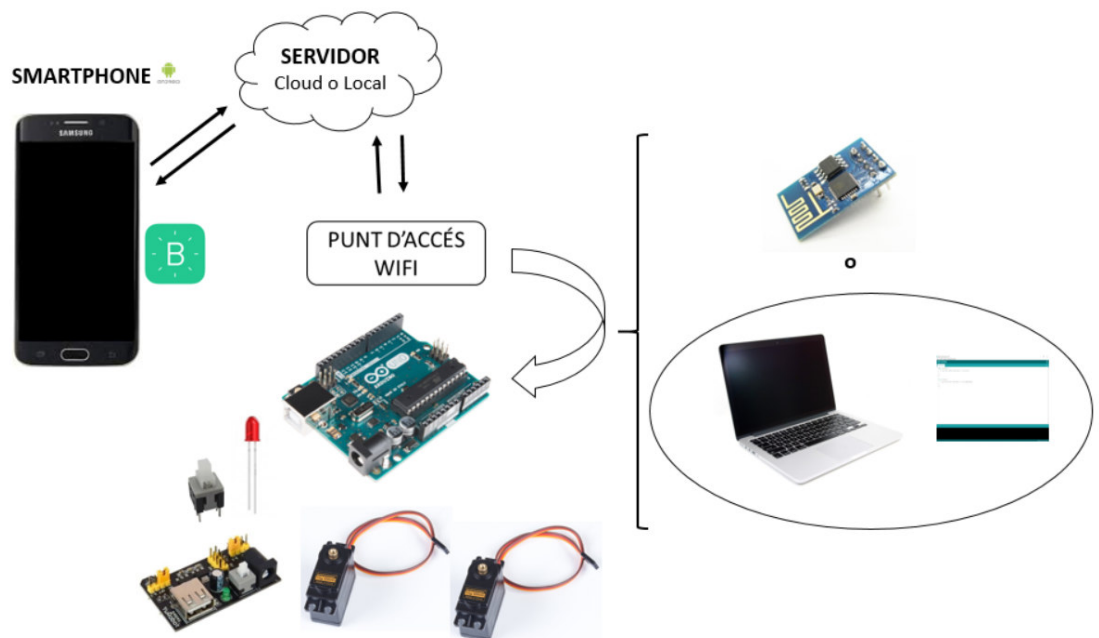


Figura 3.1. Elements del sistema desenvolupat



Tant l'smartphone com la placa Arduino han de disposar d'una connexió d'Internet, per comunicar-se a través d'un servidor. D'aquesta manera s'intercanvien informació de control i de monitorització dels servomotors.

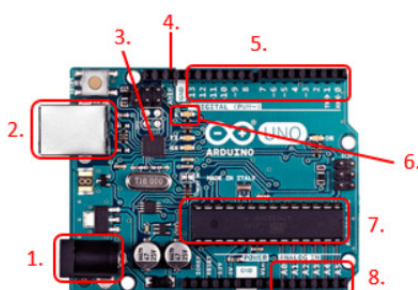
Cal destacar que l'smartphone no requereix cap connexió directa amb la placa Arduino, tipus Bluetooth o cable, ja que la comunicació s'estableix mitjançant WiFi o connexió de dades 3G/4G.

En els següents apartats s'estudien les especificacions tècniques principals dels diferents elements i s'analitzen les seves funcionalitats. També es valoren diverses opcions de configuració per obtenir un bon rendiment del sistema.

3.2. Arduino UNO R3

Arduino UNO R3 és una plataforma d'Open-Source i Open-Hardware, que és una eina molt adequada per al desenvolupament d'aquest projecte.

La placa (Fig. 3.2.) està basada en un microcontrolador principal ATmega328P (7). Inclou un altre microcontrolador ATmega16U2 (3), que està programat com a convertidor USB-Sèrie que canalitza la comunicació sèrie de RX/TX del microcontrolador principal al port USB.



Entrades i sortides digitals

- **Pin 0 (RX) i Pin 1 (TX):** entrada (RX) i sortida (TX) del port sèrie.
- **Pin 2 i 3:** entrades d'interrupcions externes.
- **Pin 3, 5, 6, 9, 10 i 11:** sortides PWM (8 bit de resolució).
- **Pin 10 (SS), Pin 11 (MOSI), Pin 12 (MISO), Pin 13 (SCK):** pins per el port de comunicació SPI.
- **Pin 13:** Aquest pin està connectat al LED.
- **Pin 4 (SDA), Pin 5 (SCL):** suporten la comunicació I²C.

Figura 3.2. Arduino UNO R3: Distribució de components i pins.

- (1)** Adaptador AC-DC **(2)** Connector USB-PC **(3)** ATmega16U2
(4) AREF: Voltatge de referència **(5)** 14 pins entrada/sortida
(6) LED pin 13 **(7)** ATmega328P **(8)** 6 entrades analògiques

La placa pot alimentar-se de dues formes diferents: mitjançant un adaptador AC-DC a través d'un connector (1) o per via USB emprant el connector (2).



Aquesta placa disposa de 14 entrades i sortides digitals (5), on cada pin realitza diferents tasques i funcionalitats. També ofereix 6 entrades analògiques (8) que es converteixen a digitals amb una resolució de 10 bits. Té la possibilitat de configurar la tensió de referència (4) per a les entrades analògiques que es pot seleccionar a través de la funció adequada emprant el software i que, per defecte, és de 5 V. També incorpora un LED (6).

Per programar i configurar l'Arduino UNO R3, es requereix instal·lar el programa Arduino IDE en un PC, i la placa ha de romandre connectada al PC mitjançant un cable USB. D'aquesta manera es pot programar el microcontrolador ATmega328P (Fig. 3.3).

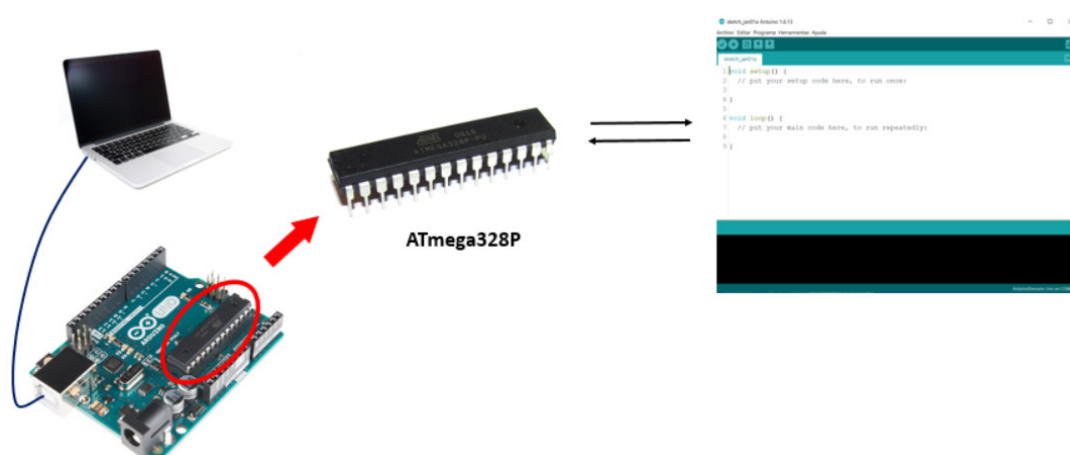


Figura 3.3. PC - Arduino UNO R3, ATmega328P i Arduino IDE

A continuació, a la taula 3.1. s'indiquen els avantatges i inconvenients de l'Arduino UNO R3.

AVANTATGES	INCONVENIENTS
Open Source.	No hi ha una normativa ni validació de funcionament aplicada en les llibreries de la xarxa.
Fàcil de programar.	Plataforma dissenyada principalment per al desenvolupament d'aplicacions.
Documentació: molta informació disponible a la xarxa.	Molta informació es troba només als fòrums on els usuaris expliquen les seves experiències.
Gran varietat de plaques.	Limitació en emprar els diferents processos al mateix temps.



Disponibilitat de shields i perifèrics.	Compatibilitat entre components no sempre és la idònia.
Baix cost.	Dificultat per l'elaboració de projectes més complexos. Per exemple, en alguns casos cal afegir una font d'alimentació externa.
Augmenta l'aprenentatge i les habilitats en l'electrònica.	-

Taula 3.1. Taula avantatges i inconvenients.

3.3. Connectivitat d'Arduino a Internet

En aquest apartat s'estudia les diferents opcions que s'han valorat i desenvolupat per obtenir la connexió a Internet. S'han considerat tres aproximacions diferents:

- La utilització d'una placa Arduino amb un mòdul WiFi integrat.
- Afegir un mòdul addicional ESP8266 a la placa Arduino UNO R3.
- Utilitzar la connexió a Internet de l'Arduino UNO R3 a través del PC connectat amb el cable USB.

3.3.1. Arduino UNO WIFI

Com a primera opció, es va considerar l'última tecnologia de hardware de la família Arduino, que és la placa Arduino UNO WIFI. Té unes especificacions i característiques molt similars a l'Arduino UNO R3, però a més a més, té integrat el mòdul de la WiFi (ESP8266).

Es va comprar la placa Arduino UNO WIFI i es van realitzar unes primeres proves per configurar la WiFi. No obstant, es va cometre l'error de no comprovar prèviament la compatibilitat amb el desenvolupador Blynk. Com que és una placa molt moderna, encara no s'havien desenvolupat les llibreries necessàries per obtenir la comunicació entre ambdós dispositius. En el fòrum d'usuaris del Blynk s'estava debatent com poder aconseguir que Blynk disposés de l'opció d'emprar-la. Degut a aquesta limitació, es va optar finalment per retornar aquest producte.



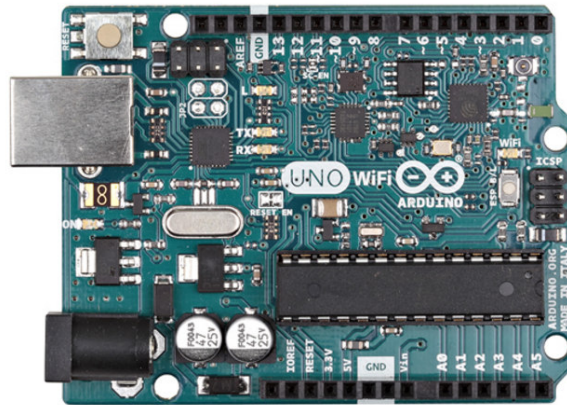


Figura 3.4. Arduino UNO WIFI

Així doncs, com que la placa no era compatible en aquell moment amb Blynk, es va tenir que comprar la placa estàndard, Arduino UNO R3, tenint present, en tot moment, que era un requisit adaptar la connexió d'aquesta placa a través d'un punt d'accés WiFi.

3.3.2. Mòdul WiFi (ESP8266)

Es va considerar una altra opció que consistia en incorporar el mòdul ESP8266 [8] connectat a la placa Arduino UNO R3. Aquest mòdul permet aconseguir un punt de connexió WIFI. Conté un xip molt potent i es pot configurar de manera que pugui obtenir una IP definida per connectar-se a Internet.

Per configurar el mòdul ESP8266 es necessita un convertidor USB TTL, per establir una connexió sèrie amb el PC, que ha de tenir el software Arduino IDE instal·lat.

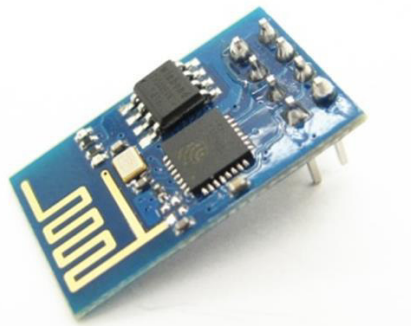


Figura 3.5. Mòdul ESP8266

Com que no es disposava d'aquest convertidor USB TTL, es va explorar alternatives per a la configuració del mòdul ESP8266.



Es va procedir a emprar l'Arduino UNO R3, de manera que si s'extreia el microcontrolador ATmega328P, la placa podia actuar com un adaptador USB-sèrie. Amb aquesta especificació, l'Arduino podia realitzar la mateixa tasca que un convertidor USB TTL, degut a què disposa d'un xip ATmega16U2 i els pins de referència TX i RX.

A partir d'aquí, es fan les connexions adequades (Fig. 3.6) i es comença a programar utilitzant les comandes AT adients. Les comandes AT són unes instruccions de codi interpretat que permeten la comunicació entre l'usuari i un terminal mòdem.

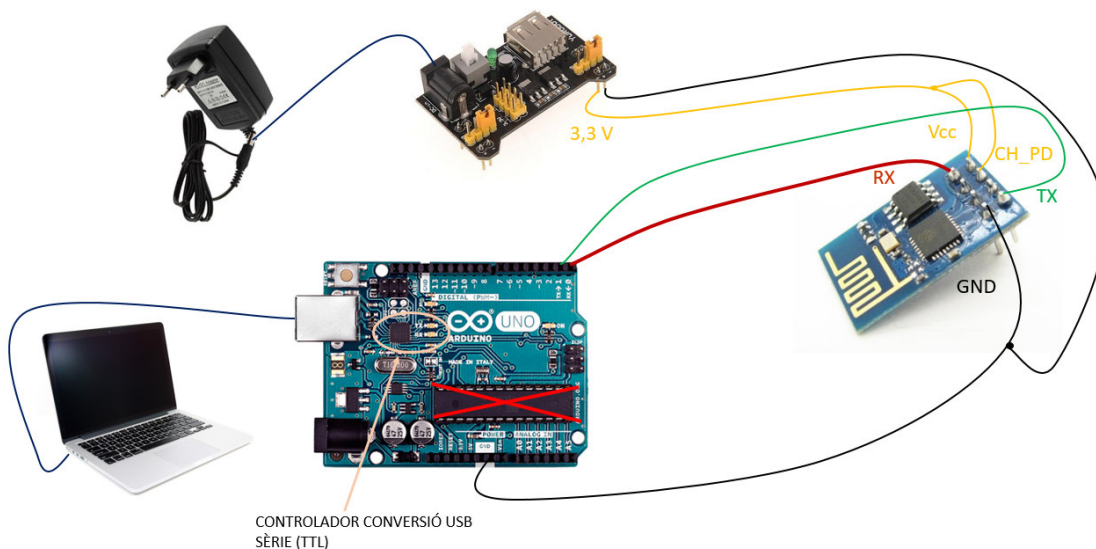


Figura 3.6. Connexió de components per configurar el mòdul ESP8266

Aquestes comandes es van realitzar a través del monitor de l'Arduino IDE, on es va definir prèviament la seva velocitat, de 9600 bauds (símbols per segon). Es van introduir les següents comandes seguint aquest ordre (Fig. 3.7) [6]:

1. **AT**, serveix per verificar si el sistema d'ordres AT està operatiu.
2. **AT+GMR**, mostra la informació sobre el firmware del mòdul.
3. **AT+CWMODE**, permet establir el mode de funcionament WiFi del mòdul. Existeix tres disponibles, (1) estació, (2) punt d'accés i (3) mixta (punt d'accés i estació). En aquest cas, s'ha de configurar de manera el mode sigui el (3).
4. **AT+CWLAP**, mostra una llista dels punts d'accés disponibles al voltant del mòdul.
5. **AT+CWJAP**, el mòdul es connecta al punt d'accés indicat amb el format de **AT+CWJAP="nom de la xarxa", "contrasenya"**.



6. **AT+CIPMUX**, estableix el mode de connexió simple o múltiple segons el format utilitzat "AT+CIPMUX=0" o "AT+CIPMUX=1" respectivament.
7. **AT+CIPSERVER**, activa o desactiva el mode del servidor segons el format utilitzat "AT+CIPSERVER=0" (que es requerirà un reinici per fer-se efectiu) o "AT+CIPSERVER=1,{port}" que començarà a acceptar peticions en el port corresponent.
8. **AT+CIFSR**, mostra la direcció IP local i actual del mòdul.
9. **AT+CIPSTA**, estableix la direcció IP de l'estació "AT+CIPSTA".

A partir d'aquí, el mòdul ESP8266 ja té la direcció IP carregada.

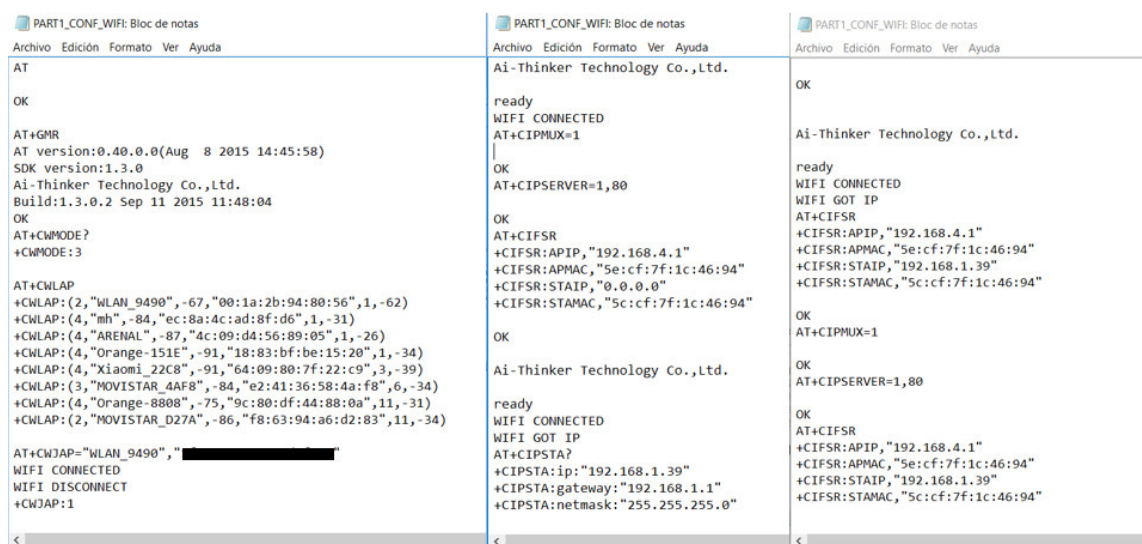


Figura 3.7. Comandes realitzades per la configuració de ESP8266

No obstant, cada vegada que es realitza un RESET, en el monitor apareix "Ai-Thinker Technology Co.,Ltd." on s'ha de tornar a escriure les comandes (AT+CIPMUX, AT+CIPSERVER i AT+CIFSR), ja que la configuració prèvia no queda guardada.

Per tant, un cop carregada la direcció IP, es va realitzar una prova per validar el seu funcionament (Fig. 3.8). Consistia en obrir un navegador d'Internet i introduir la direcció IP. Es va escriure "/provaWifi", i a la finestra del monitor es va confirmar aquesta crida al mòdul ESP8266 des del navegador. D'aquesta manera, es va verificar la correcta configuració del mòdul WiFi.



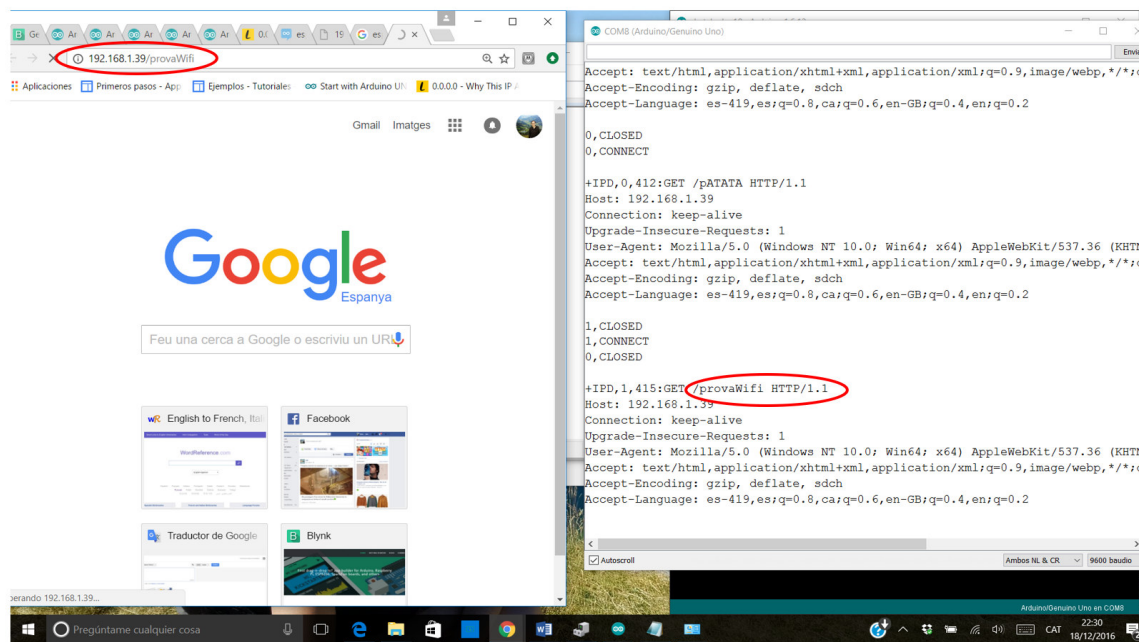


Figura 3.8. Validació de la configuració del mòdul ESP8266

De totes maneres, cada vegada que es volia emprar aquest mòdul després d'un RESET, es necessitava ser reconfigurat de nou. Això fa que sigui necessària que la connexió USB TTL estigui activa i calgui escriure tres comandes AT interpretades cada vegada que es vol utilitzar el mòdul WiFi.

En conclusió, degut a aquestes limitacions del mòdul ESP8266, es considera que no és una opció viable i efectiva. Per això, es va optar per cercar una altra opció per obtenir la connexió a Internet.

3.3.3. Cable USB connectat al PC

Després de valorar les dues aproximacions, i comprovar les seves limitacions. Es decideix buscar una altra solució per obtenir la connexió de placa Arduino a Internet.

Així doncs, s'opta per emprar la connexió de l'Arduino UNO R3, mitjançant un cable USB, a un PC que facilitarà l'accés a Internet. D'aquesta manera, es pot desenvolupar totes les funcionalitats del sistema que es proposen en aquest projecte (Fig. 3.9).



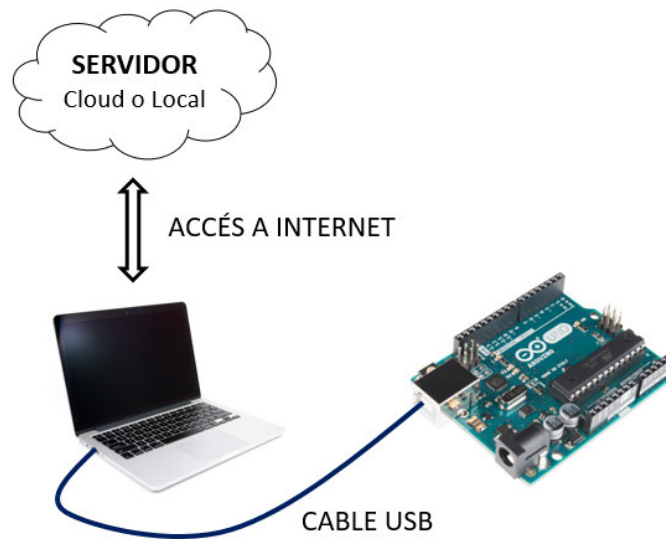


Figura 3.9. Configuració final d'accés a Internet de la placa Arduino UNO R3

A continuació, des de l'Arduino IDE es va configurar la placa, definint la velocitat de connexió sèrie a 9600 bauds, i codificant les eines necessàries per disposar de la connexió d'Internet a través del PC. En el següent capítol, a l'apartat 4.2 s'explicarà en detall totes les eines de programació i llibreries requerides per a la realització d'aquest projecte.

3.4. Servomotors SunFounder

S'ha optat per utilitzar dos servomotors de la marca SunFounder [12]. Aquest servomotor d'engranatge de metall permet rotar des de 0 fins a 180 graus i té un parell motor de 13,5 kg. Es pot controlar el seu tren d'engranatges per poder girar i mantenir una posició determinada. Les seves característiques permeten que sigui controlat des d'una placa Arduino. El cable de connexió és de 30 cm de llarg (Fig. 3.10), on el cable taronja és el cable del senyal de control, el cable vermell és el de l'alimentació i el cable marró és el de terra (GND). El servomotor treballa a una tensió compresa entre 4,8 V i 6 V. La seva dimensió és de 54,1 x 20 x 44,3 mm i té un pes molt lleuger de 55 g.



Figura 3.10. Servomotor SunFounder



Els cables de senyal del servomotor estan connectats als pins de sortida PWM de l'Arduino, que estan codificats amb una resolució de 8 bits. Aquests pins corresponen als numerats com 3, 5, 6, 9 i 11.

Quan el servomotor rep un tren de polsos (Fig. 3.11), segons l'amplada d'aquests es determina la posició angular del servomotor. L'amplada varia entre 0,5 ms a 2,5 ms i la freqüència a la que es treballa és de 50 Hz (20 ms).

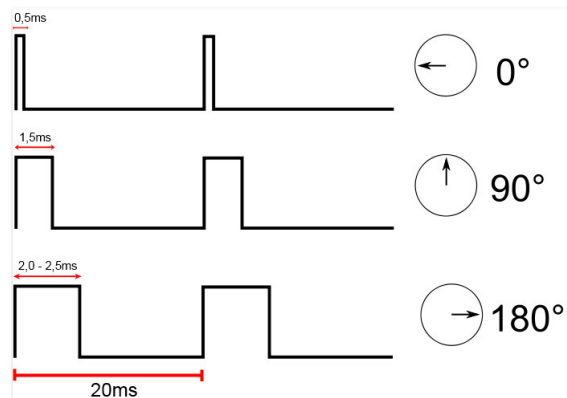


Figura 3.11. Tren de polsos

Existeixen diverses llibreries que faciliten el control dels servomotors des de la plataforma Arduino. En l'apartat 4.2 es descriuen en detall les seves característiques.

3.4.1. Font d'alimentació YwRobot

Els servomotors necessiten ser alimentats amb una tensió compresa de 4.8 V a 6 V. Es podria utilitzar la font d'alimentació de 5 V que disposa l'Arduino. Però es descarta aquesta opció degut a què es fa servir dos servomotors, perquè la placa Arduino no pot subministrar suficient corrent per alimentar-los. Per tant, s'ha considerat oportú fer servir una font d'alimentació externa per assegurar un bon funcionament i una bona engegada. S'ha escollit aquesta placa YwRobot que disposa de dos canals d'alimentació, cadascú configurable a 3,3 V o 5 V.

Cal recordar què quan es fa servir la font d'alimentació, la massa d'aquesta s'ha de connectar a la massa de la placa Arduino per tenir la mateixa referència a tot el sistema.



Figura 3.12. Font d'alimentació YwRobot



La font d'alimentació obté l'energia que subministra de la xarxa elèctrica a través d'un adaptador que li proporciona una tensió contínua compresa entre 6,5 V i 12 V. El corrent màxim que subministra és de 700 mA, però si es fa servir el connector USB del PC, llavors és aproximadament de 400 mA.

3.5. Sensor de presència

Es vol considerar un sensor de presència al prototipus demostrador, de manera que el control de les posicions s'activi només quan hi ha algú assegut i retorni a les posicions inicials, quan no hi ha ningú a la butaca.

Com una primera aproximació, se simularà aquest sensor utilitzant un interruptor. Quan l'interruptor estigui en mode ON, representarà que l'usuari està assegut a la butaca. En cas contrari, indicarà que no hi ha ningú a la butaca. En el prototipus s'implementa un LED que serà una indicació de la presència de l'usuari.

El circuit de l'interruptor (Fig. 3.13) requereix d'una resistència de 330 Ω que es connecta al pin de 5 V de l'Arduino i a un dels terminals de l'interruptor, que també està connectat al pin 2 de l'Arduino. L'altre terminal de l'interruptor es connecta a massa (GND). Per indicar visualment que hi ha algú assegut s'ha afegit un LED vermell a l'Arduino, connectant el seu càtode al pin 13 de l'Arduino i el seu ànode a massa (GND).

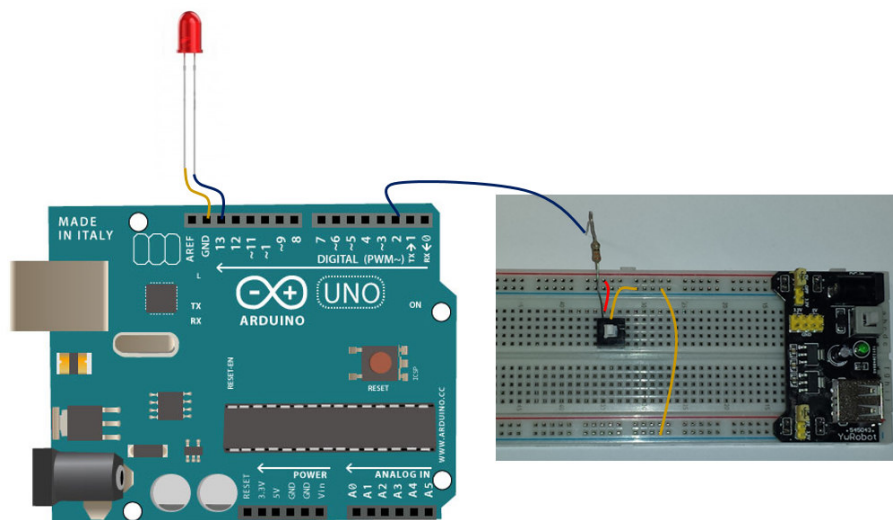


Figura 3.13. Sensor de presència



3.6. Descripció del sistema conjunt

En aquest apartat s'explica la connexió de tots els components del sistema comentats anteriorment (Fig. 3.14).

Els cables de senyal dels dos servomotors es connecten als pins 6 i 9 de l'Arduino, degut a què són pins de sortida PWM. Gràcies a aquesta connexió es pot controlar els servomotors des de la placa Arduino, a través del software Arduino IDE. Els altres dos pins dels servomotors es connecten a cada canal de la font d'alimentació externa, un a 5 V i l'altre a massa, perquè el funcionament dels dos servomotors siguin correctes. L'Arduino és alimentat i configurat des del PC.

Un cop s'ha explicat els components escollits i la seva interconnexió, es mostra el seu conjunt amb tots els elements integrats. En el següent capítol s'estudia les eines de programació i llibreries per configurar tots aquests components físics.

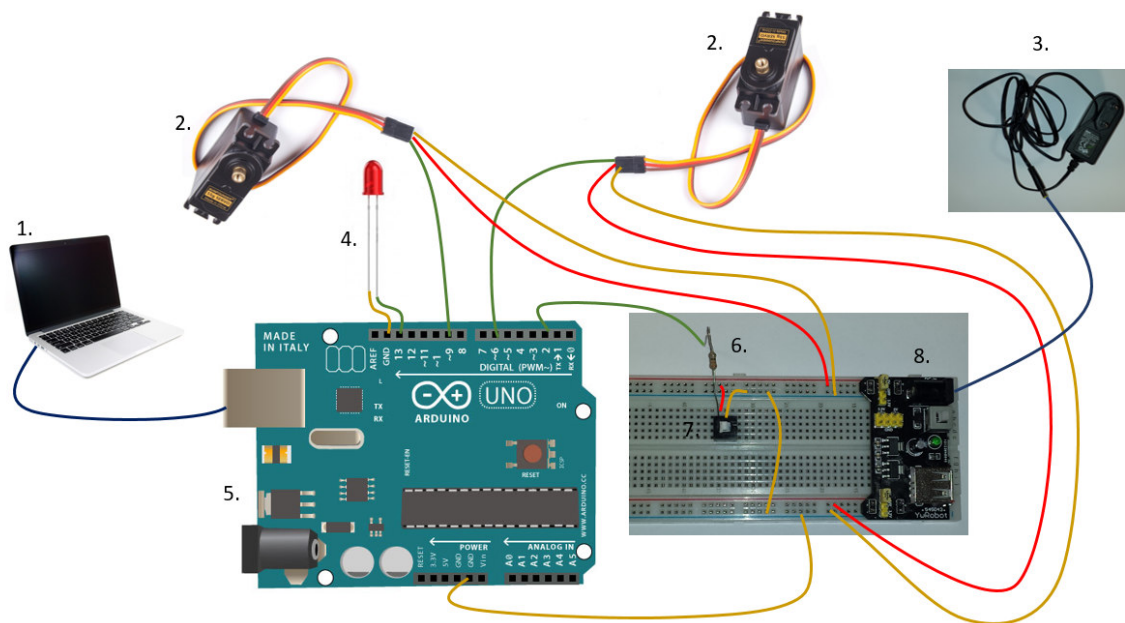


Figura 3.14. Esquema de la connexió dels components:

1. PC
2. Servomotors
3. Adaptador
4. LED
5. Arduino UNO R3
6. Resistència de 330 Ω
7. Interruptor
8. Font d'alimentació de 5 V



4. Eines de programació i llibreries

En l'apartat anterior s'ha comentat els components hardware i com es connectava cada dispositiu. Aquest capítol es centra en la metodologia de les eines de programació de Blynk, Arduino i les seves llibreries.

4.1. Eines de Blynk

Per establir la comunicació entre l'Arduino i l'smartphone amb l'App Blynk, es requereix descarregar la llibreria Blynk al PC i configurar els dispositius seleccionats. Aquesta llibreria s'obté des de la pàgina oficial de Blynk i permetrà la connexió dels dispositius amb el servidor i processar totes les ordres entrants i sortints. Així doncs, amb els components configurats i les seves llibreries adequades, permetran comunicar-se amb el servidor de Blynk, tant l'smartphone com la placa Arduino (Fig. 4.1).

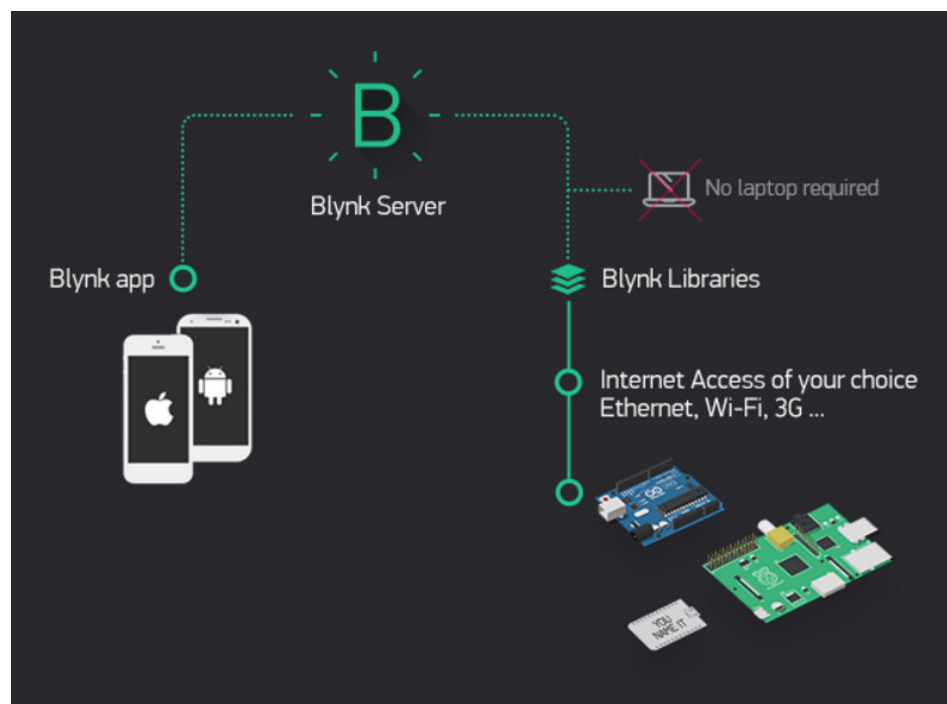


Figura 4.1. Esquema connectivitat (Extreta de la pàgina web oficial de Blynk)

Les aplicacions de Blynk estan dissenyades per a IoT, que permeten controlar el hardware remotament, mostrar les dades dels sensors, emmagatzemar les dades i visualitzar-les.



Blynk pot controlar pins analògics i digitals d'entrada i sortida de l'Arduino directament. També permet configurar uns pins virtuals que serveixen perquè l'usuari pugui enviar i rebre dades del microcontrolador a l'smartphone.

Els pins virtuals són una eina molt útil, ja que poden facilitar que els diferents widgets del Blynk puguin intercanviar informació. Per exemple, en el cas d'un servomotor connectat a la placa Arduino, els widgets Slider Settings, Labeled Value Settings o Gauge Settings poden enviar i/o visualitzar la posició del servomotor.

Les eines del desenvolupador Blynk ofereixen sis blocs diferents de widgets:

- Controllers: Button (de tipus pulsadors o interruptor), Slider, Timer, Joystick, zeRGBa, Step H, Step V.
- Displays: Value Display, Labeled Value, LED, Gauge, LCD, Graph, History Graph, Terminal, Video Streaming, Level H, Level V.
- Interface: Tabs, Menu, Table, Time Input, Map, Device Selector.
- Notifications: Twitter, Notification, Email.
- Others: Bridge, Real-time clock, BLE, Bluetooth, Eventor, Music Player, WebHook.
- Smartphone Sensors: GPS Stream, GPS Trigger, Accelerometer, Gravity Sensor, Light Sensor, Proximity Sensor, Temperature Sensor, Humidity Sensor, Barometer Sensor.

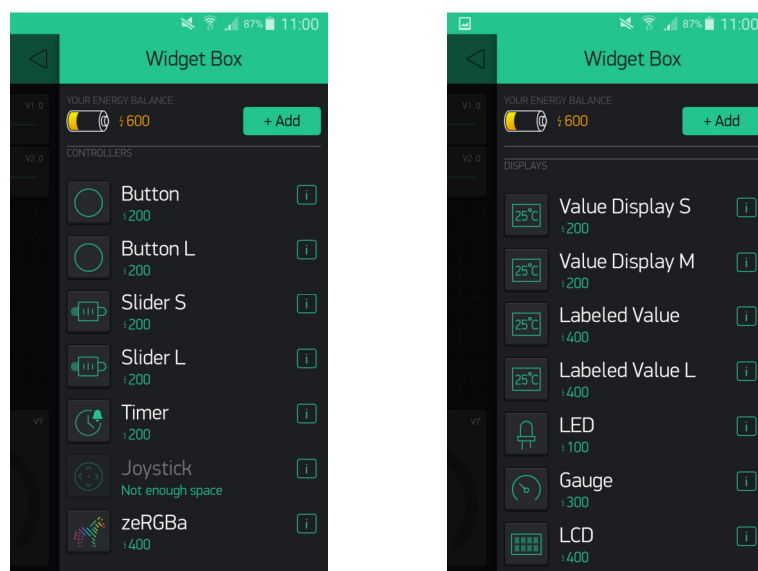


Figura 4.2. Exemples de widgets disponibles al Blynk



Una de les característiques més importants d'aquests widgets és que poden configurar, seleccionant els pins reals o virtuals i els seus paràmetres de control, i incorporar directament al tauler de l'aplicació.

Blynk ofereix un nombre limitat de widgets gratuïts. El fabricant de Blynk utilitza la terminologia "*Energy Balance*" que indica l'energia "consumida" per utilitzar cada widget. Quan es crea un compte de Blynk, regalen una energia total de 2000 i quan se supera aquest límit, requereix contractar un servei de pagament per obtenir-ne més. Cada widget té assignat un determinat cost ("*Energy Balance*"). Per exemple, el Button té un cost de 200. Quan s'elimina un widget, el cost serà retornat.

L'aplicació dissenyada per aquest projecte s'ha realitzat escollint uns widgets determinats dins dels límits de l'aplicació gratuïta que ofereix Blynk.

4.2. Arduino IDE

La placa Arduino es programa a través de l'entorn de programació, Arduino IDE. Permet escriure programes, compilar-los i carregar els executables a la plataforma hardware. Està preparat per treballar en diversos sistemes operatius com Windows, Mac OS X i Linux. L'entorn de programació està escrit en Java i està basat en *Processing* i altres software d'Open Source. Aquest software es pot utilitzar per a qualsevol placa de la família Arduino.

La versió de l'Arduino IDE que s'ha utilitzat en aquest projecte és la 1.6.13, que va ser la última versió que es va descarregar en aquell moment. No obstant, les versions s'actualitzen ràpidament.

Arduino IDE disposa de diferents llibreries integrades, com ArduinoCloud, Bridge, EEPROM, Esplora, Ethernet, Firmata, HID, Keyboard, Mouse, Robot Control, Robot IR Remote, Robot Motor, SD, SPI, Software Serial, SpacebrewYun, Wire. També es poden incorporar llibreries externes ("*contributed libraries*"), com Blynk, BlynkESP8266_Lib, GSM, LiquidCrystal, Servo, SimpleTimer, Stepper, TFT, Time, TinyGSM, VarSpeedServo-master, WiFi. Algunes llibreries ja estaven implementades, però d'altres necessàries per aquest projecte es van tenir que buscar en webs d'altres desenvolupadors, i es van introduir manualment a l'IDE.



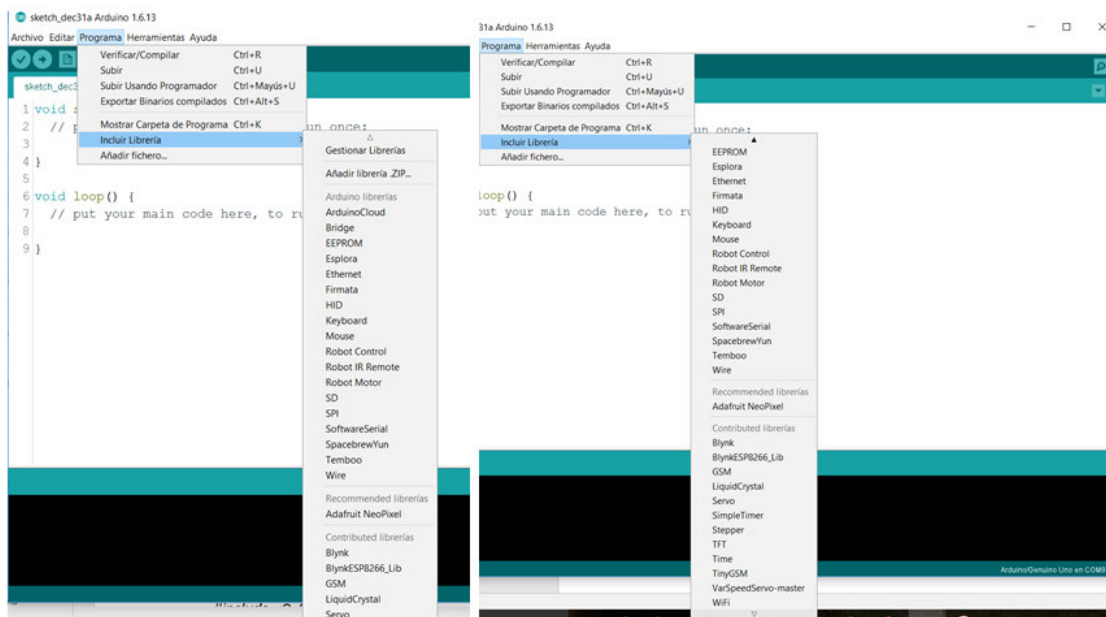


Figura 4.3. Llistat de llibreries a l'Arduino IDE

S'ha analitzat les llibreries disponibles per Arduino i Blynk. Per al desenvolupament d'aquest projecte s'han inclòs tres llibreries específiques:

- `#include <SoftwareSerial.h>`
- `#include <VarSpeedServo.h>`
- `#include <BlynkSimpleStream.h>`

Arduino té incorporat en hardware el suport per a la comunicació sèrie a través dels pins 0 i 1. La comunicació es fa des d'un UART, que està integrat al xip ATmega328P.

La llibreria “*SoftwareSerial.h*” permet desenvolupar la comunicació sèrie en altres pins digitals d'Arduino. D'aquesta manera és possible tenir múltiples ports sèrie, que cal configurar per software, fins a una velocitat de 115200 bps (bit per segon).

Per al control dels servomotors, inicialment es va estudiar les possibilitats que ofereix la llibreria “*Servo.h*”, que és la que està disponible a l'Arduino IDE. Es va observar que és una llibreria molt limitada en quant als paràmetres de control i, per tant, no era l'adequada per a aquest projecte. Per això es van buscar altres alternatives. Finalment es va trobar una llibreria més completa, “*VarSpeedServo.h*”, que incorpora paràmetres de control de velocitat.



Aquesta llibreria permet realitzar una sèrie de tasques, com assignar el nom de cada servomotor, indicar a quin pin d'Arduino es connecta, dir a quin angle es vol arribar i seleccionar la velocitat.

La llibreria “*BlynkSimpleStream.h*” és la llibreria idònia per al cas de què no es tingui cap shield WiFi de la família Arduino i l'Arduino UNO R3 no tingui cap connectivitat a Internet. Amb aquesta llibreria s'habilita la connexió Arduino - PC i permet que la placa rebí dades d'Internet a través del PC. Per tant, amb aquesta llibreria es pot aconseguir que Blynk es comuniqui amb l'Arduino a través del USB connectat amb el PC.

Un cop es té les llibreries adients, per escriure el codi es procedeix a crear una nova aplicació (“*sketch*”, segons la terminologia de l'Arduino IDE), com es mostra en la següent Fig. 4.4, que permet introduir i programar el codi adequat per aquest projecte.

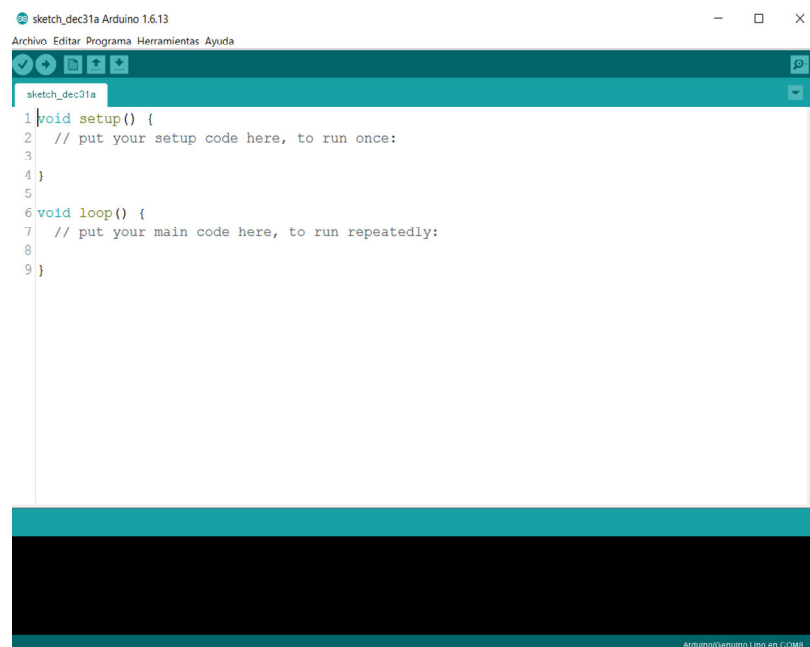


Figura 4.4. Nou sketch

Es crea el sketch amb dos blocs definits. El primer bloc del programa és el *setup()*, on hi ha el codi d'inicialització i que només s'executa una vegada. En l'altre bloc hi ha el *loop()*, que és on es posa el codi principal que s'executa repetidament.

Tot seguit, s'indica el tipus de placa amb el que s'està treballant i amb quin port es treballa per poder establir la comunicació entre el PC i l'Arduino. En aquest cas, es treballa amb la placa “Arduino/Genuino UNO” i amb el port “COM8” (Fig. 4.5).



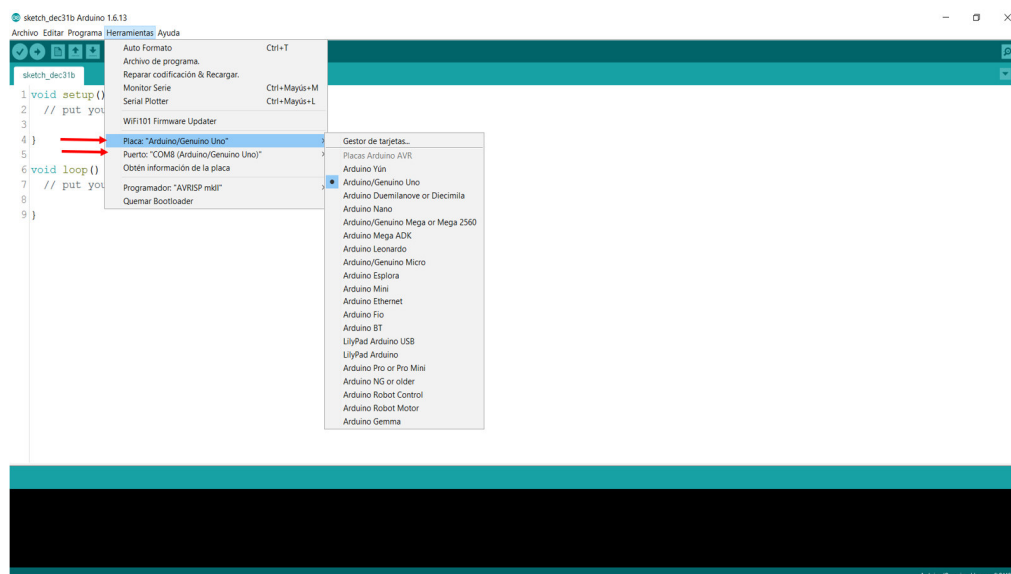


Figura 4.5. Assignació placa i port

4.3. Metodologia per crear un projecte amb Blynk per Arduino

El projecte creat en el desenvolupador Blynk es comunicarà amb la placa Arduino a través del núvol (Cloud Server) o un servidor local de Blynk (Fig. 4.1). Abans d'establir aquesta connexió cal seguir una sèrie de passos. Primer de tot es descarrega l'aplicació Blynk al mòbil Android, on es crea un usuari de Blynk indicant un correu electrònic de referència. A continuació s'instal·len les llibreries de Blynk al PC, però si no es té instal·lat el software Arduino IDE en el PC, és imprescindible descarregar-lo primer i instal·lar-lo.

Una vegada instal·lat tot, se segueix per establir la connexió entre l'aplicació Blynk i el programa Arduino IDE. Des d'aplicació Blynk es crea el projecte que es vol dur a terme, introduint el hardware que s'utilitzarà (la placa de control) i com serà la connexió (Ethernet, WiFi, USB, etc.).

Un cop assignat i creat el projecte, Blynk envia un correu a l'usuari de manera automàtica, en el que s'indica el link de l'AuthToken, que és l'identificador de seguretat que permet buscar aquest projecte creat al núvol de Blynk, per establir la comunicació entre el mòbil i Arduino. Aquest link s'ha d'incorporar al programa de l'Arduino IDE.

Inicialment, per defecte, en l'aplicació es considera la utilització del Cloud Server de Blynk, on l'AuthToken permet accedir al núvol de Blynk des de l'smarphone i la placa Arduino.

A continuació es mostra un exemple de creació d'un nou projecte amb Blynk (Fig. 4.6). S'assigna un nom de l'aplicació ("ProvaDeFuncionament"), es defineix la placa de hardware



seleccionada ("Arduino UNO") i el tipus de connexió a Internet de la placa seleccionada ("USB").

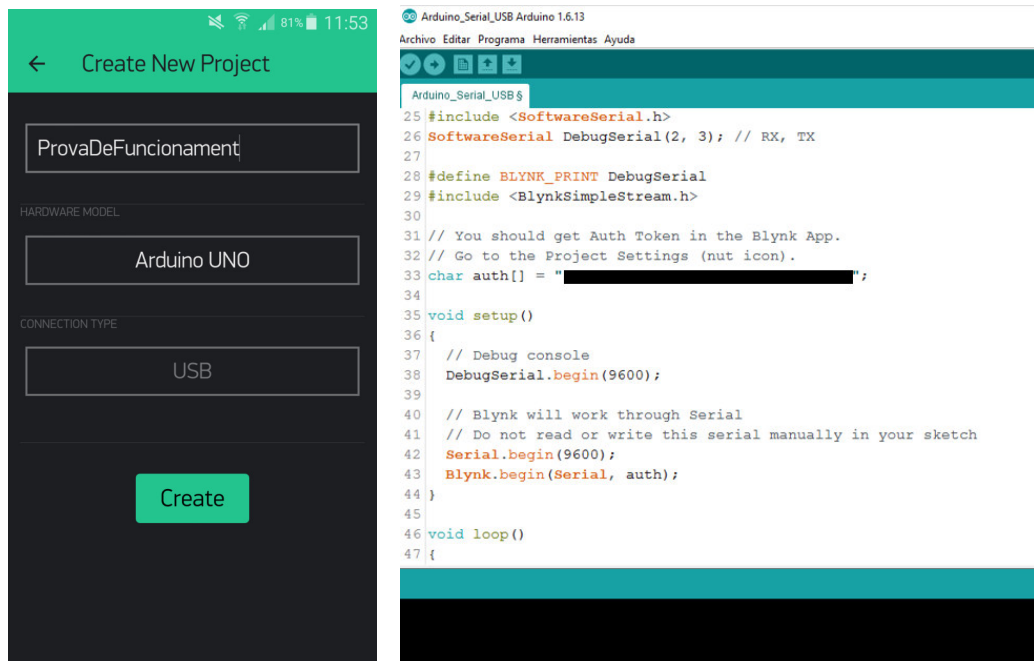


Figura 4.6. Exemple Blynk i codi de programa en l'Arduino IDE

Per executar el programa, primer cal verificar el codi creat en l'Arduino IDE, compilar-lo i carregar-lo a la placa Arduino.

Seguidament, s'obre la finestra de l'administrador del PC, on s'escriurà la comanda (Fig. 4.7):

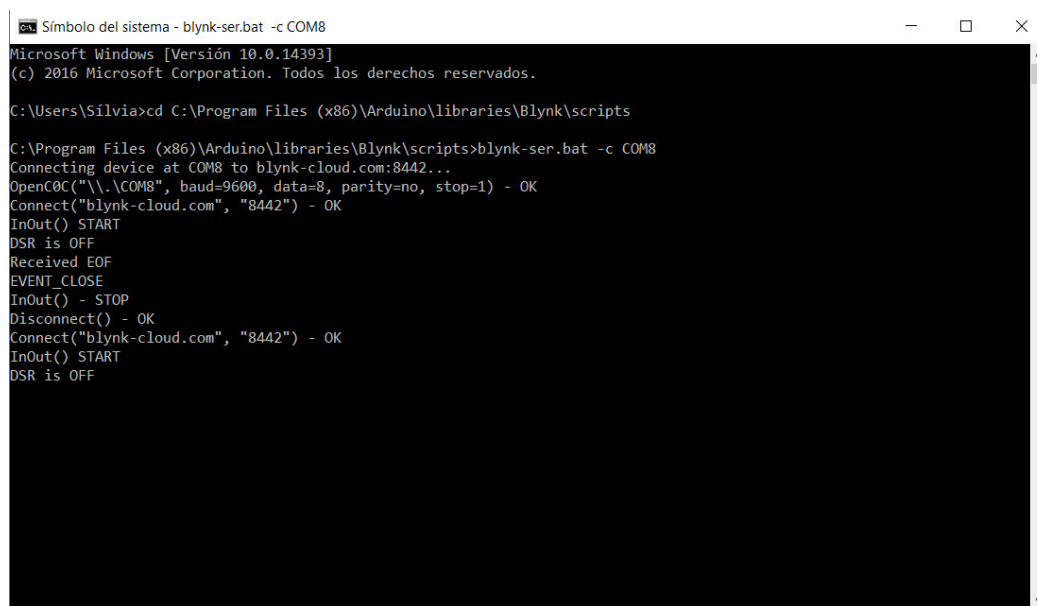
```
cd C:\Program Files (x86)\Arduino\libraries\Blynk\scripts
```

que és el path dels arxius per executar els programes de Blynk. A continuació, s'executa la comanda:

```
blynk-ser.bat -c COM8
```

on hi ha una sèrie de comandes que s'executaran una sola vegada i que, a més a més, assigna el port amb el que l'Arduino UNO es comunicarà amb l'Arduino IDE (en aquest cas, el port COM8).





```
Símbolo del sistema - blynk-ser.bat -c COM8
Microsoft Windows [Versión 10.0.14393]
(c) 2016 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Sílvia>cd C:\Program Files (x86)\Arduino\libraries\Blynk\scripts

C:\Program Files (x86)\Arduino\libraries\Blynk\scripts>blynk-ser.bat -c COM8
Connecting device at COM8 to blynk-cloud.com:8442...
OpenCOM8("\\.\\COM8", baud=9600, data=8, parity=no, stop=1) - OK
Connect("blynk-cloud.com", "8442") - OK
InOut() START
DSR is OFF
Received EOF
EVENT_CLOSE
InOut() - STOP
Disconnect() - OK
Connect("blynk-cloud.com", "8442") - OK
InOut() START
DSR is OFF
```

Figura 4.7. Finestra de l'administrador del PC

Un cop realitzades aquestes comandes (Fig. 4.7), tal com es pot veure a la finestra de l'administrador (CMD admin), la connexió entre el Blynk i el PC està habilitada, a través del “*blynk-cloud.com*”. Per tant, es pot començar a treballar amb el projecte configurant i manipulant els widgets que ofereix Blynk.

4.4. Exemple de prova de funcionament de Blynk

Abans de desenvolupar l'aplicació proposada en aquest treball, que es presentarà al capítol 5, es fa prèviament una prova senzilla per comprovar la metodologia per crear un projecte, i verificar que la connexió és correcta.

La prova de funcionament consisteix en encendre i apagar un LED des de l'smartphone, utilitzant un widget “*Button*” creat amb el desenvolupador Blynk. La informació s'envia a través del CloudServer del Blynk a la placa Arduino, que està connectada al PC. Per altra banda, es col·loca el pin càtode del LED al pin 13 de l'Arduino i el pin ànode al pin GND de l'Arduino.

Cal recordar que el pin 13 de l'Arduino, té implementada la interfície per connectar directament un LED, i per tant no és necessari afegir més components.

S'agafa una aplicació d'exemple del Blynk (Fig. 4.8), que permet habilitar la connexió amb l'Arduino a través del PC mitjançant la connexió USB.



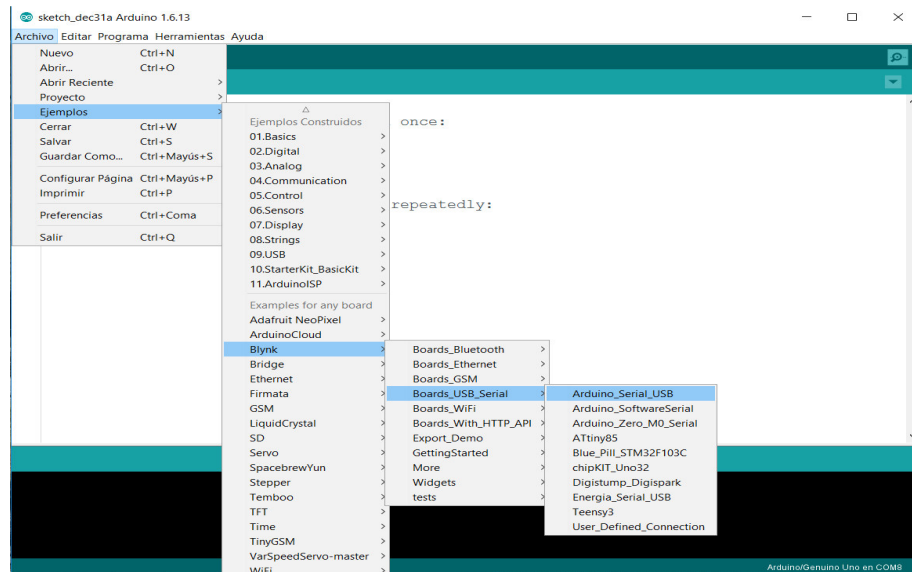


Figura 4.8. Exemple Arduino_Serial_USB

Tot seguit, en el codi de l'exemple seleccionat, s'introdueix el link de l'AuthToken per habilitar la connexió entre PC-Arduino-smartphone Android. Es verifica i es compila en la plataforma del desenvolupament del software Arduino IDE. A continuació, s'envien les comandes adients a la finestra de l'administrador del PC (Fig. 4.9).

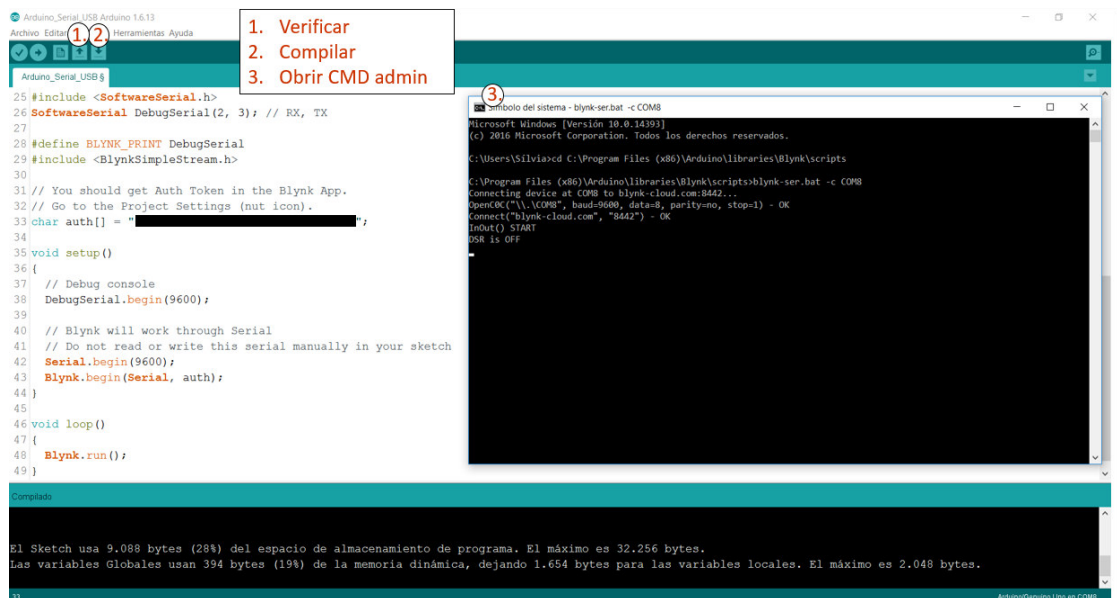


Figura 4.9. Passos per executar el codi



Ara, amb aquest codi ja s'ha activat la comunicació entre l'Arduino i Blynk. Es procedeix a introduir el widget en el tauler del projecte "ProvaDeFuncionament" en l'aplicació de Blynk. S'indica el pin D13, on està connectat el LED (Fig. 4.10).

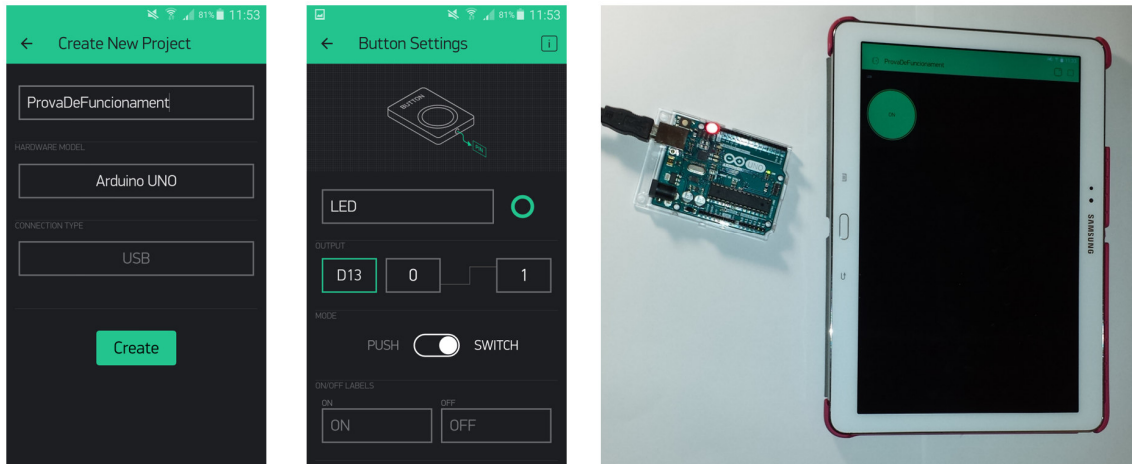


Figura 4.10. Procediment creació projecte ProvaDeFuncionament

Ara ja es pot manipular aquest switch, s'observa que quan s'activa l'interruptor de l'smartphone, el LED connectat a l'Arduino s'encén (Fig. 4.10). Per tant, s'ha aconseguit habilitar aquesta comunicació a través del CloudServer.

Amb aquesta prova de funcionament s'ha comprovat el bon funcionament entre el hardware i el software de tots els dispositius.

Cal destacar que aquesta aplicació "ProvaDeFuncionament" es pot utilitzar des de qualsevol dispositiu Android, smartphone o tablet.



5. Disseny i desenvolupament del sistema

5.1. Disseny i desenvolupament de l'aplicació

En aquest capítol es descriu en detall el disseny del prototipus demostrador realitzat en el projecte. Com ja s'ha comentat en els capítols anteriors, es configura la connexió entre tots els dispositius i es procedeix a elaborar el codi del programa en les plataformes Blynk i Arduino, per aconseguir els objectius proposats.

Es descriuen els elements escollits i la seva configuració, per a l'aplicació de mòbil ("widgets" de Blynk). També es detallen les funcions seleccionades de les llibreries de Blynk, Arduino i servomotors, per a la seva comunicació i control.

El sistema dissenyat en aquest projecte inclourà quatre tipus de widgets a l'aplicació del mòbil, per aconseguir les funcions proposades:

- Widget "*Slider*", per moure un servomotor a la posició desitjada.
- Widget "*Gauge*", per visualitzar la posició angular del servomotor.
- Widget "*Button*", per guardar una posició angular, recuperar una posició guardada o realitzar un moviment cap a les posicions màximes dels dos servomotors.
- Widget "*Terminal*", per visualitzar les variables assignades. És una eina orientada per a la depuració ("*debug*") i verificació del bon funcionament del programa.

D'altra banda, cal destacar les funcions escollides de les llibreries esmentades anteriorment:

- Control dels servomotors: assignant el nom de cada un d'ells, indicant a quin pin de la placa Arduino es connecta i a quin angle es mou amb la seva velocitat assignada.
- Control des de l'aplicació Blynk: on s'introduirà el link de l'AuthToken, i es realitzarà una sèrie de comandes per poder activar el funcionament i control dels widgets seleccionats. Aquests widgets estaran assignats als diferents pins virtuals (V1, V2, V3, etc.), de manera que es faciliti la programació en el codi i la informació que s'intercanvia. També permetran enviar i rebre informació per al control de la butaca.



S'han estudiat les diferents parts del programa per poder fer front a les condicions de funcionament del sistema. En cada bloc del programa s'analitza com utilitzar les funcions de les llibreries seleccionades.

A continuació, es detalla el disseny de l'aplicació proposada en aquest projecte (Fig. 5.1). En aquest disseny s'ha inclòs un sensor de presència, que fa de clau d'activació del sistema de control als servomotors. Quan està actiu el sistema, disposa d'una sèrie d'opcions de control de la butaca, només quan hi ha algú assegut.

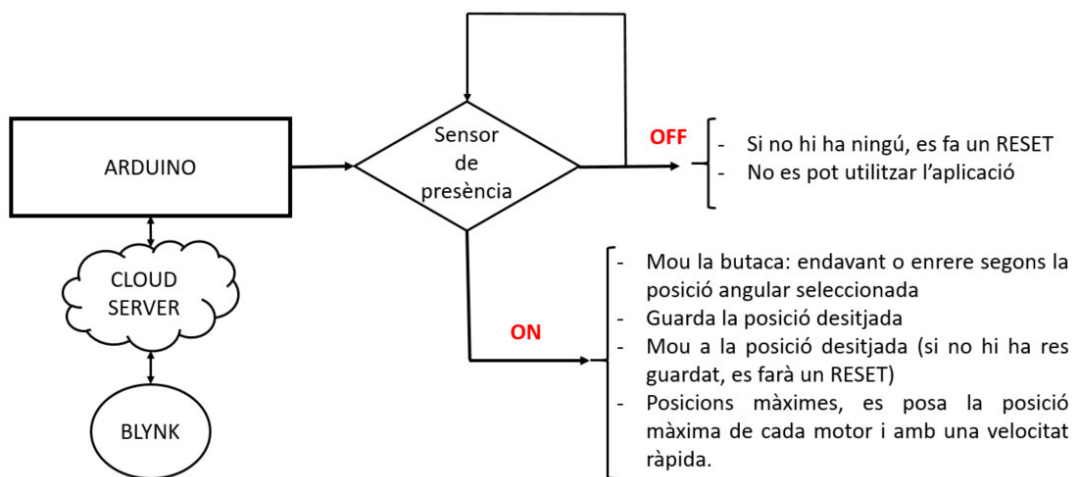


Figura 5.1. Esquema de funcionament del sistema

Un cop estan connectats tots els dispositius del sistema, tal com s'ha indicat a la figura 3.14, es comença a programar les funcions proposades anteriorment.

Es procedeix a configurar el codi de manera que quan no hi ha ningú assegut, la butaca retorni a la posició inicial i no es pugui controlar des de l'aplicació encara que l'usuari la intenti fer servir. En el prototipus demostrador es col·loca un interruptor al protoboard, que representarà un sensor mecànic de presència (Fig. 5.2).



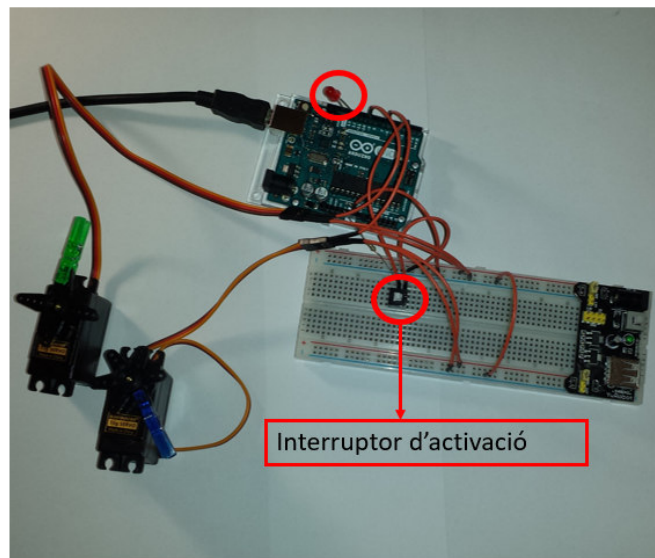


Figura 5.2. Circuit amb l'interruptor de presència, la placa Arduino i els servomotors

A continuació, es programen les següents opcions de control de la butaca, per a quan hi ha algú assegut.

Es vol aconseguir controlar els servomotors de forma individualitzada. Per això es defineix una variable per a cada un d'ells. En aquest projecte se li assigna a través de la comanda "VarSpeedServo" amb els següents noms:

VarSpeedServo reposapeus;

VarSpeedServo respatller;

Seguidament, dins de la funció "void setup()", s'indica a quins pins es connecten els servomotors a la placa Arduino, a través de la funció "nomServo.attach(pin)".

La funció "nomServo.write(angle, speed, wait)" permet seleccionar la posició angular desitjada. On cal introduir l'angle que es vol obtenir, comprés entre 0 i 180°. La velocitat ("speed") està codificada en 8 bits, per això pot prendre un valor enter entre 0 i 255 (on la velocitat màxima correspon al valor 255). El paràmetre "wait" és un booleà que assenyala, quan és "True", que el programa ha d'esperar fins que el servomotor arribi a la posició indicada, abans d'executar la següent instrucció. Si és "False", s'activa aquesta comanda, però es procedeix immediatament a executar la següent sentència, encara que el servomotor estigui en moviment.

La selecció de la posició ("angle") de cada servomotor es farà a través d'un "Slider" de l'aplicació Blynk. En el widget s'imposarà els límits de les posicions angulars de cada servomotor, en un rang de 0 a 180°. En aquesta aplicació s'ha configurat el moviment del



reposapeus en un rang $[0^\circ, 80^\circ]$ i el respatller en un rang $[0^\circ, 30^\circ]$, i s'ha vinculat als pins virtuals V1 i V2 respectivament (Fig. 5.3).

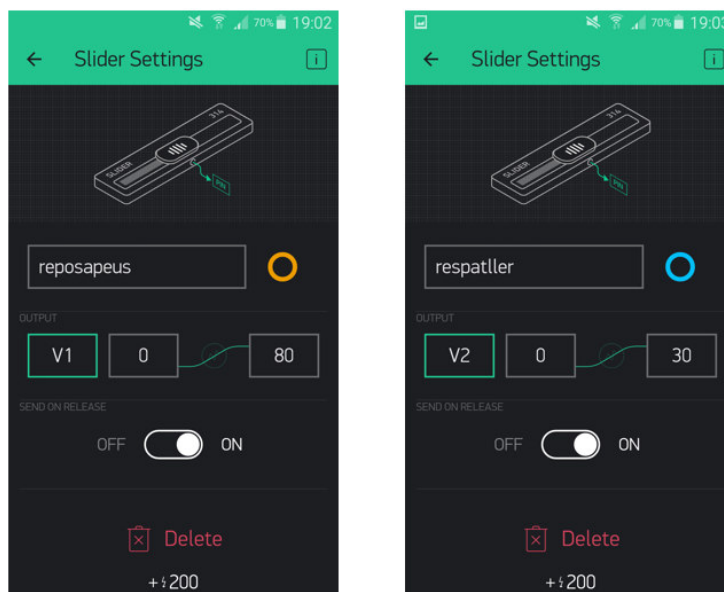


Figura 5.3. Configuració dels “Sliders” de cada motor

En el codi s'introduirà la funció “*BLYNK_WRITE(pin Virtual)*” assignant quin pin virtual s'utilitza en el widget i cridant la comanda “*param.asInt()*”, perquè llegeixi el valor que pren el “Slider”. D'aquesta manera amb aquesta configuració i la funció:

“*nomServo.write(angle, speed, bool)*”

s'ordena al motor que es mogui a la posició desitjada (Fig. 5.4). S'ha seleccionat una velocitat predefinida de valor enter igual a 5 (“*vel_1*” i “*vel_2*”), que és una velocitat lenta, pensant en el millor confort de l'usuari.

```

124 BLYNK_WRITE(V1) {
125   value1 = param.asInt();
126   reposapeus.write(value1, vel_1, true);
127   Blynk.virtualWrite(V5,value1);
128 }
129 BLYNK_WRITE(V2) {
130   value2 = param.asInt();
131   respatller.write(value2, vel_2, true);
132   Blynk.virtualWrite(V7,value2);
133 }

```

Figura 5.4. Codi de programació dels Sliders de cada motor



Per visualitzar el seu valor angular, es pot emprar els dos widgets “*Labeled Value Settings*” o “*Gauge Settings*”. Degut a la limitació de cost i d'espai en el tauler d'instruccions, només s'utilitza un dels dos. S'ha escollit el “*Gauge Settings*” ja que visualitza millor que l'altre widget. Com s'ha explicat abans, dins de la funció “*BLYNK_WRITE(pinVirtual)*” es posa la comanda “*Blynk.virtualWrite(pinVirtual, valor)*” de manera que aquests widgets mostrin la posició en angle, seleccionats a través del “*Slider*” (Fig. 5.4). En aquests widgets també s'imposen els límits dels angles d'ambdós servomotors (Fig. 5.5).

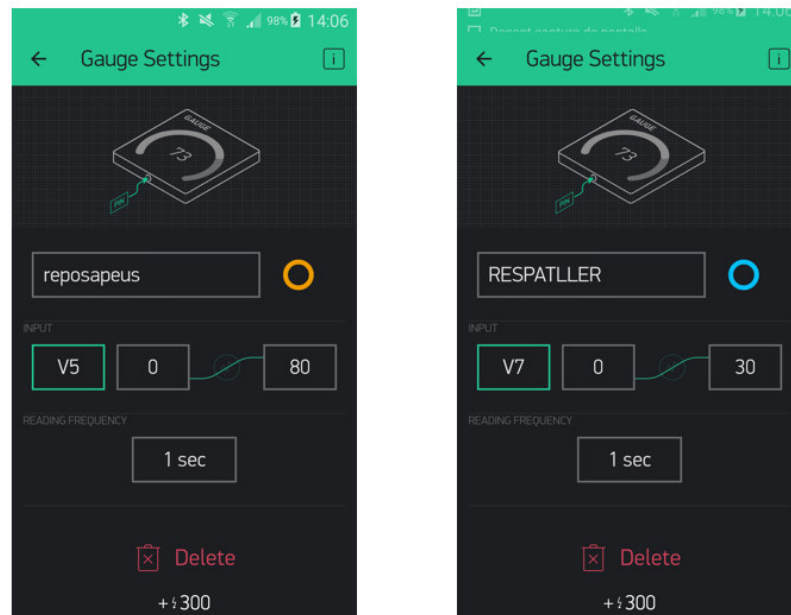


Figura 5.5. Configuració dels “Gauge” de cada motor

Una altra condició addicional per al sistema de control de la butaca motoritzada, és que l'usuari tingui la prestació de guardar la posició desitjada per utilitzar-la en un altre moment, sempre i quan s'hagi mantingut alimentat el sistema.

Per exemple, si l'usuari està gaudint de la butaca en la posició desitjada, té la possibilitat de guardar-la polsant al widget anomenat “Save”. Si s'aixeca per algun motiu, la butaca retornarà a la posició inicial, però l'usuari disposarà de l'opció de polsar el botó “Move”, perquè li mogui a la posició que estava fa uns minuts, només si abans l'ha guardat amb l'opció “Save”.

Aquestes funcions es faran a través del widget “*Button Settings*”, que es configura de manera que la sortida només pugui prendre el valor 0 i 1 i el seu mode sigui “*Push*” (polsador). Cal indicar a quin pin virtual s'ha emprat, el widget “Save” es té vinculat al pin V10 i el “Move” al pin V11 (Fig. 5.6).



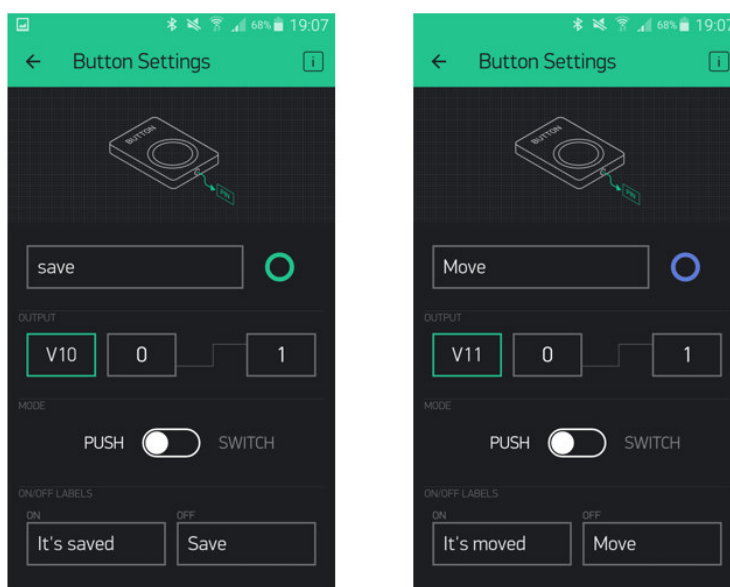


Figura 5.6. Configuració dels “Button Settings” del SAVE i MOVE

L'última prestació disponible per a l'usuari permet posar les posicions màximes de cada motor directament. Es tracta d'un widget “Button Settings” que, quan l'usuari premi aquest polsador, mourà la butaca per arribar a les posicions màximes (Fig. 5.7). D'aquesta manera l'usuari s'estalvia de fer servir el “Slider” i arriba a la posició de manera ràpida i directe.



Figura 5.7. Configuració del “Button Settings” de POS MÀXIMES



En aquesta opció s'ha programat un sistema de control dels dos servomotors, de manera que el moviment es faci en tres trams amb diferents velocitats preestablertes. La velocitat variable del widget “Pos Màximes”, està pensada pel millor confort de l'usuari, fent que l'inici i el final del moviment es realitzi amb unes velocitats més lentes. Es defineixen tres trams de velocitats respecte al moviment total: el primer tram fins al 15%, el segon tram fins al 75% i l'últim fins al 100%. A cada tram s'assignarà una velocitat determinada (Fig. 5.8).

En el cas del reposapeus, el primer tram es farà amb una velocitat de valor enter igual a 5, el segon amb una velocitat igual a 40 i l'últim tram a una velocitat lenta de valor 10. El mateix procediment es fa en el cas del respalller, però en el segon tram amb una velocitat de 30.

```

74  if(PosMax==1){
75      terminal.print("Posicions Maximes");
76      //reposapeus
77      INC1=value1_max - value1;
78      terminal.print("INC1=");terminal.println(INC1);
79      value1=value1+(INC1*0.15);
80      reposapeus.write(value1, 5, true);
81      terminal.print("s'ha mogut fins a, 1er tram=");terminal.println(value1);
82      terminal.print("velocitat, 1er tram=");terminal.println(5);
83      value1=value1+(INC1*0.60); //incrementa fins a 75%
84      reposapeus.write(value1,40,true);
85      terminal.print("s'ha mogut fins a, 2n tram=");terminal.println(value1);
86      terminal.print("velocitat, 2n tram=");terminal.println(40);
87      reposapeus.write(value1_max, 10, true);
88      terminal.print("s'ha mogut fins a, 3r tram=");terminal.println(value1_max);
89      terminal.print("velocitat, 3r tram=");terminal.println(10);
90      Blynk.virtualWrite(V1, 80);
91      Blynk.virtualWrite(V5, 80);

```

Figura 5.8. Codi de programació del “Button Settings” de POS MÀXIMES (reposapeus)

Un cop configurades les diverses condicions, a continuació es descriu el programa per al cas que no hi ha ningú assegut. Es tracta que quan es desactiva l'interruptor manualment, la butaca realitzi un RESET automàtic i els widgets “Slider” i “Gauge Settings” retornin a les posicions inicials. A més a més, es configura de manera que si el sensor està desactivat, l'usuari no pugui manipular l'aplicació. Els detalls d'aquest bloc es poden veure a l'apartat 5.3.

Per al desenvolupament del codi de l'aplicació s'ha considerat un widget addicional, per facilitar la depuració del programa. S'ha incorporat el widget “Terminal Settings” (Fig. 5.9), amb l'objectiu que retorni tots els valors de les variables i les comandes executades. Aquest widget s'ha vinculat al pin virtual V6. És un widget de referència que només s'utilitzarà per al desenvolupament del prototipus demostrador.



En la realitat, a l'usuari no l'interessa veure aquest widget en el tauler d'instruments ja que està pensat per a la programació. Per això, en l'aplicació final i definitiva per l'usuari, no s'incorporarà al tauler.

En el codi s'utilitza la comanda "`terminal.println(value1)`", de manera que es pugui visualitzar els valors assignats a la pantalla del mòbil, en diversos punts del programa. Quan es vol enviar un string es fa a través de la comanda "`terminal.print("Escriu el que vulguis")`".

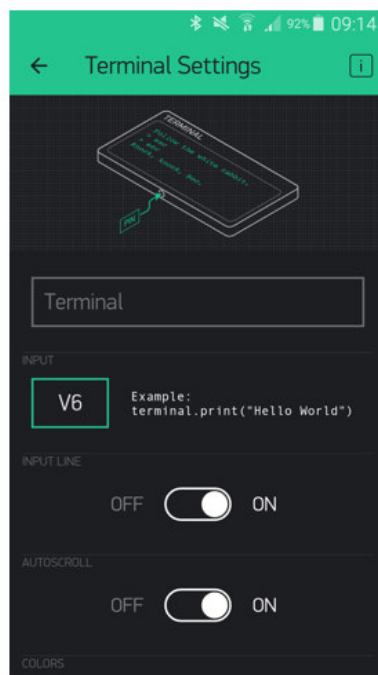


Figura 5.9. Configuració del "Terminal Settings"

5.2. Descripció de l'aplicació MA_BUTACA a l'entorn BLYNK

Per anomenar l'aplicació s'ha volgut reconèixer la gran utilitat d'emprar els dos dispositius: mòbil i Arduino. Per això l'App es diu MA_BUTACA ("Mobile-Aduino BUTACA") on es reconeix el fet d'aconseguir la connexió entre aquests dos dispositius, la seva configuració i el seu paper per poder controlar el que es proposa en aquest projecte.

També es destaca en aquest nom, el fet que MA_BUTACA sigui una abreviatura de la "meva butaca", ja que aquesta aplicació està dirigida per a tot tipus d'usuari que vulguin seleccionar la seva butaca a la posició desitjada i de confort.



5.2.1. Aplicació MA_BUTACA per al desenvolupament

En l'apartat anterior s'ha explicat el disseny de l'aplicació i l'ús de cada widget. En aquest apartat, es comenta breument l'aplicació realitzada per al desenvolupament del prototipus utilitzant el widget “*Terminal*”. Aquest widget ha sigut una eina molt important per poder visualitzar cada un dels paràmetres durant l'execució del programa, comprovar si es realitzava correctament les seves funcionalitats i depurar el programa.

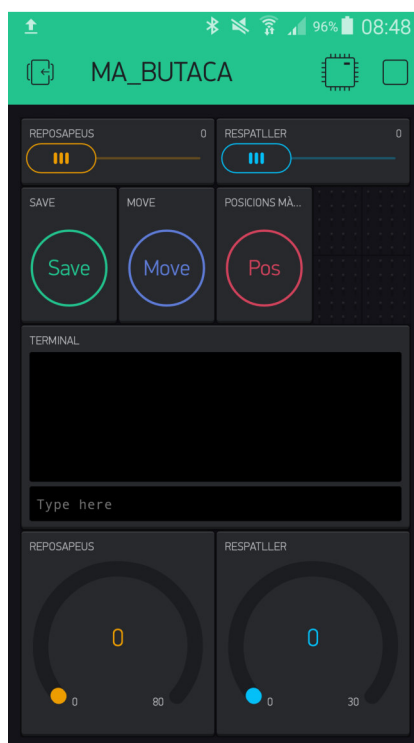


Figura 5.10. Tauler d'instruments de l'app MA_BUTACA per al desenvolupament

Durant la primera fase de la programació s'han trobat una sèrie de dificultats per poder desenvolupar de manera adequada i coherent el codi, degut el desconeixement de la programació del Blynk i del llenguatge C/C++. No obstant, ha permès aprendre la seva metodologia i el seu ús adequat.

Per altra banda, el fet de ser aplicacions d'Open-Source fa que hi hagin moltes llibreries fetes per diferents autors, que no sempre funcionen correctament, no estan ben documentades o no són adequades per als dispositius del projecte (Blynk, Arduino i Servomotors).

Un dels problemes més concrets que es va trobar durant la depuració del programa, va ser la lectura de les posicions que prenen els “*Sliders*”, quan es polsava el botó “*SAVE*”. Es va observar que al “*Terminal*” no retornava la posició assignada. La causa va ser que no



estaven ben configurades les variables globals. D'aquesta manera, es va poder corregir aquest error.

Per solucionar els diferents problemes del programa, també es va procedir a buscar diferents exemples d'ús de les funcions del Blynk. Així es van detectar els errors principals que es feien. Finalment, es va poder millorar el programa i estructurar-lo de manera adient.

En conclusió, amb l'ajuda del widget “*Terminal*” es va poder comprovar i verificar que el programa responia bé a les condicions imposades sense realitzar cap error.

5.2.2. Aplicació MA_BUTACA per l'usuari

En aquest apartat s'explica l'aplicació definitiva i dirigida a l'usuari. En l'aplicació es treu el widget “*Terminal*” degut a què a l'usuari no l'interessa comprovar si la codificació funciona bé.

Així doncs, el fet de prescindir del widget “*Terminal*” permet disposar de més espai en el tauler d'instruments. Per tant, es decideix emprar els widgets esmentats anteriorment amb unes mides més grans, per facilitar i fer que el seu ús sigui més còmode per a l'usuari.

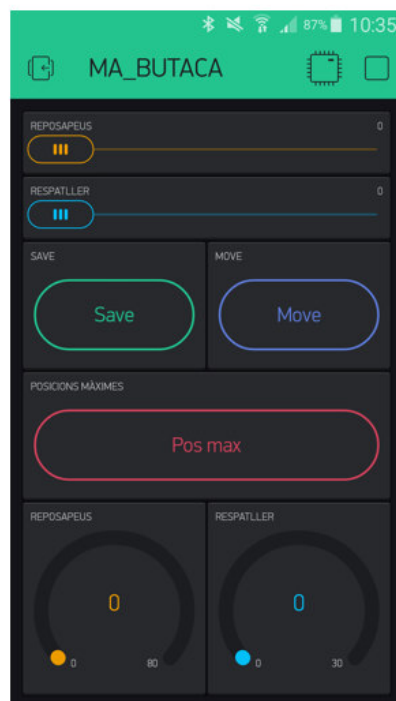


Figura 5.11. Tauler d'instruments de l'app MA_BUTACA per a l'usuari



5.3. Codi de funcionament de MA_BUTACA a l'Arduino

A continuació s'exposa el codi que s'ha creat per controlar la butaca motoritzada que es proposa en aquest projecte amb les condicions indicades anteriorment.

```
prova_proposta_TFG Arduino 1.6.13
Archivo Editar Programa Herramientas Ayuda

prova_proposta_TFG
1 #include <SoftwareSerial.h>
2
3
4 SoftwareSerial DebugSerial(2,3); //RX,TX
5
6 #include <VarSpeedServo.h>
7
8 #define BLYNK_PRINT DebugSerial
9 #include <BlynkSimpleStream.h>
10
11 char auth[] = " ";
12
13 VarSpeedServo reposapeus;
14 VarSpeedServo respatller;
15
16 WidgetTerminal terminal(V6);
17
18 int value1 = 0;
19 int value2 = 0;
20 int value1_max=80;
21 int INC1;
22 int value2_max=30;
23 int INC2;
24 int vel_1=5;
25 int vel_2=5;
26
27
28 int SavePos1;
29 int SavePos2;
30
31 int SavePos;
32 int MovePos;
33
34
35 int PosMax;
36 int d=2;
37
38 void setup() {
39   pinMode(2, INPUT);
40   pinMode(13, OUTPUT);
41   //Debug console
42   DebugSerial.begin(9600);
43
44   //Blynk will work through Serial
45   Serial.begin(9600);
46   Blynk.begin(auth, Serial);
47   reposapeus.attach(9);
48   respatller.attach(6);
49 }
```



```

50
51 void loop() {
52   d=digitalRead(2);
53   if (d==0) {
54     digitalWrite(13,HIGH);
55     Blynk.run();
56     if (SavePos==1) {
57       SavePos1=value1;
58       SavePos2=value2;
59       terminal.print("He guardat ja");
60       terminal.print("SavePos1 =");terminal.println(SavePos1);
61       terminal.print("SavePos2 =");terminal.println(SavePos2);
62     }
63     if (MovePos==1) {
64       reposapeus.write(SavePos1,vel_1,true);
65       terminal.print("Pos1=");terminal.println(SavePos1);
66       respatller.write(SavePos2,vel_2,true);
67       terminal.print("Pos2=");terminal.println(SavePos2);
68       Blynk.virtualWrite(V1, SavePos1);
69       Blynk.virtualWrite(V2, SavePos2);
70       Blynk.virtualWrite(V5, SavePos1);
71       Blynk.virtualWrite(V7, SavePos2);
72       SavePos=0;
73     }
74     if (PosMax==1) {
75       terminal.print("Posicions Maximes");
76       //reposapeus
77
78       INC1=value1_max - value1;
79       terminal.print("INC1=");terminal.println(INC1);
80       value1=value1+(INC1*0.15);
81       reposapeus.write(value1, 5, true);
82       terminal.print("s'ha mogut fins a, 1er tram=");terminal.println(value1);
83       terminal.print("velocitat, 1er tram=");terminal.println(5);
84       value1=value1+(INC1*0.60); //incrementa fins a 75%
85       reposapeus.write(value1,40,true);
86       terminal.print("s'ha mogut fins a,2n tram=");terminal.println(value1);
87       terminal.print("velocitat,2n tram=");terminal.println(40);
88       reposapeus.write(value1_max, 10, true);
89       terminal.print("s'ha mogut fins a, 3r tram=");terminal.println(value1_max);
90       terminal.print("velocitat, 3r tram=");terminal.println(10);
91       Blynk.virtualWrite(V1, 80);
92       Blynk.virtualWrite(V5, 80);
93       //respatller
94       INC2=value2_max - value2;
95       terminal.print("INC2=");terminal.println(INC2);
96       value2=value2+(INC2*0.15);
97       respatller.write(value2, 5, true);
98       terminal.print("s'ha mogut fins a, 1er tram=");terminal.println(value2);
99       terminal.print("velocitat, 1er tram=");terminal.println(5);
100      value2=value2+(INC2*0.60); //incrementa fins a 75%
101      respatller.write(value2,30,true);
102      terminal.print("s'ha mogut fins a,2n tram=");terminal.println(value2);
103      terminal.print("velocitat,2n tram=");terminal.println(40);
104      respatller.write(value2_max, 10, true);
105      terminal.print("s'ha mogut fins a, 3r tram=");terminal.println(value2_max);
106      terminal.print("velocitat, 3r tram=");terminal.println(10);
107      Blynk.virtualWrite(V2, 30);
108      Blynk.virtualWrite(V7, 30);
109      //
110      INC1=0;
111      INC2=0;

```



```

111     }
112 }
113 else{
114     digitalWrite(13,LOW);
115     reposapeus.write(0,5, true);
116     Blynk.virtualWrite(V1,0);
117     Blynk.virtualWrite(V5,0);
118     respatller.write(0,5, true);
119     Blynk.virtualWrite(V2,0);
120     Blynk.virtualWrite(V7,0);
121 }
122 }
123
124 BLYNK_WRITE(V1){
125     value1 = param.asInt();
126     reposapeus.write(value1, vel_1, true);
127     Blynk.virtualWrite(V5,value1);
128 }
129 BLYNK_WRITE(V2){
130     value2 = param.asInt();
131     respatller.write(value2, vel_2, true);
132     Blynk.virtualWrite(V7,value2);
133 }
134 BLYNK_WRITE(V10){
135     SavePos= param.asInt();
136 }
137 BLYNK_WRITE(V11){
138     MovePos=param.asInt();
139 }
140 BLYNK_WRITE(V12){
141     PosMax=param.asInt();
142 }
143
144
145
146

```

Figura 5.12. Codi de programació de l'aplicació MA_BUTACA

5.4. Exemple de funcionament del sistema

Les diferents condicions emprades en l'aplicació MA_BUTACA que s'han explicat en els anteriors apartats s'exposa a continuació en imatges, per visualitzar de forma senzilla les funcions de les diferents opcions que ofereix aquesta aplicació.

A continuació s'explica un exemple de com podria ser l'ordre d'ús de l'aplicació MA_BUTACA, quan l'usuari s'asseu a la butaca i manipula els següents widgets:



1. Sensor de presència activat: L'usuari s'asseu.



Figura 5.13. Exemple de funcionament pas 1

2. Moure els Sliders: L'usuari selecciona la posició desitjada, els motors es mouen a aquestes posicions. L'aplicació mostra gràficament l'angle de la posició de cada motor.

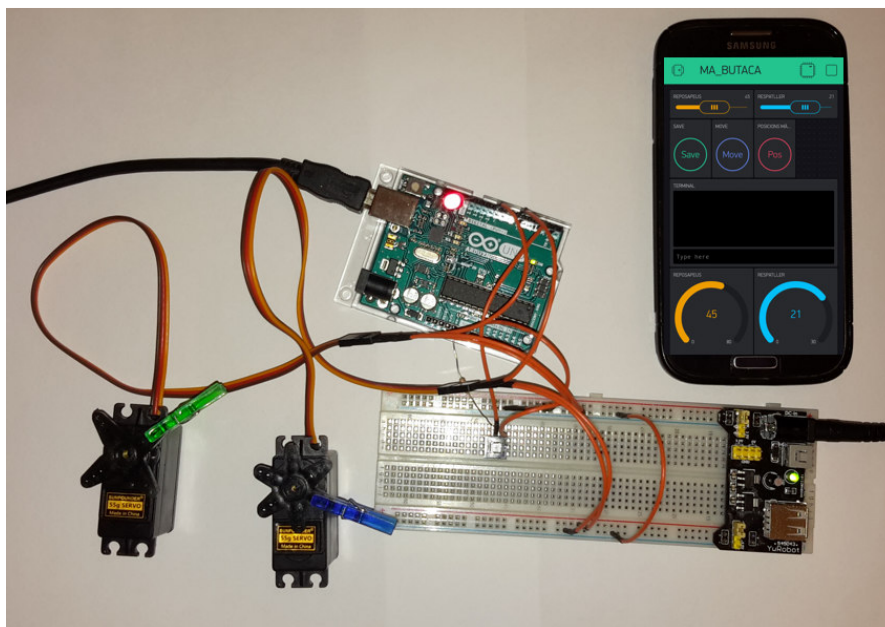


Figura 5.14. Exemple de funcionament pas 2



5. Sensor de presència activat: L'usuari torna.



Figura 5.17. Exemple de funcionament pas 5

6. Posar les posicions màximes: L'usuari selecciona les posicions màximes.

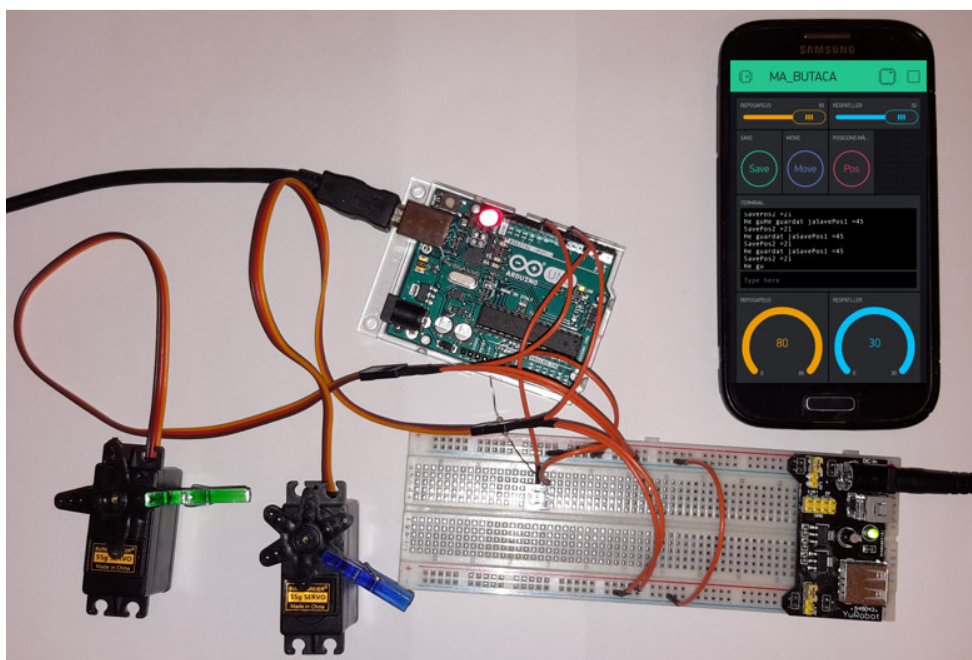


Figura 5.18. Exemple de funcionament pas 6



7. Mostra la posició guardada: L'usuari vol tornar a estar a la posició desitjada que ha guardat anteriorment (pas 3).



Figura 5.19. Exemple de funcionament pas 7





6. Manual d'utilització de MA_BUTACA

Es crea un manual d'utilització dirigit a tot tipus d'usuari que vulgui utilitzar la butaca.

Quan l'usuari desitja seure a la butaca, haurà d'obrir l'aplicació MA_BUTACA des del Blynk de manera que pugui manipular-la. En el moment en què l'usuari s'assegui, s'activarà el sensor mecànic i ja podrà fer ús de MA_BUTACA.

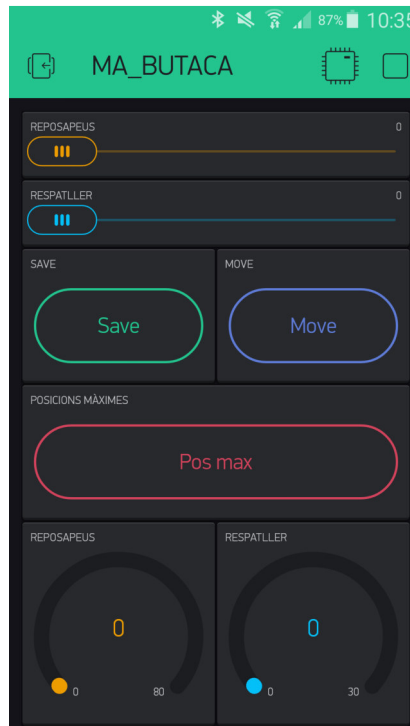


Figura 6.1. App MA_BUTACA

A continuació, s'explica la funcionalitat de cada widget.

1. LLISCADOR REPOSAPEUS

Permet controlar la posició del reposapeus que ofereix de 0° a 80°.

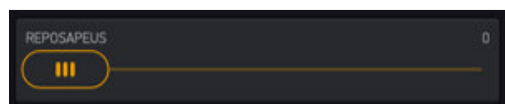


Figura 6.2. Lliscador REPOSAPEUS



2. LLISCADOR RESPATLLER

Permet controlar la posició del respatller entre 0 i 30°.

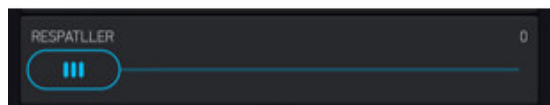


Figura 6.3. Lliscador RESPATLLER

3. BOTÓ SAVE

Permet guardar la posició actual per l'usuari.

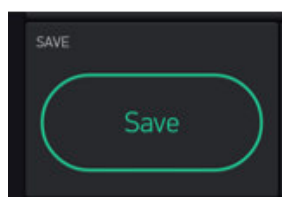


Figura 6.4. Botó SAVE

4. BOTÓ SHOW

Permet posar la posició que ha guardat anteriorment l'usuari.

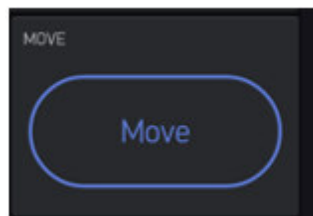


Figura 6.5. Botó MOVE

5. BOTÓ POS MAX

Permet arribar a les posicions màximes del reposapeus i del respatller de manera directa i ràpida.

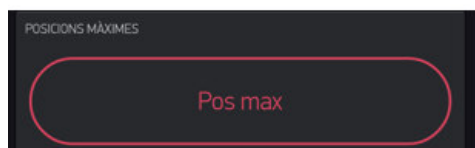


Figura 6.6. Botó POS MAX



6. VISUALITZADOR REPOSAPEUS

És un visualitzador que permet saber a quina posició està el reposapeus.

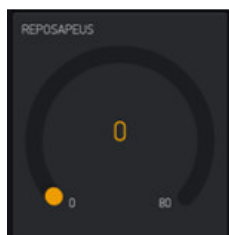


Figura 6.7. Visualitzador REPOSAPEUS

7. VISUALITZADOR RESPATLLER

És un visualitzador que permet saber a quina posició està el respatlle.

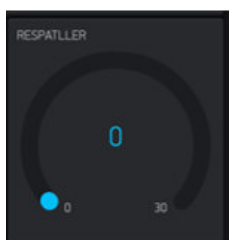


Figura 6.8. Visualitzador RESPATLLER





7. Pressupost

A continuació es desglossa els costos d'aquest projecte en dos categories diferents: els costos directes, que engloben el cost de recursos materials i de recursos humans on s'especifica la dedicació de treball, i els costos indirectes.

En aquest apartat no es té en compte el pressupost per a una possible comercialització degut a que aquest projecte es basa en demostrar la viabilitat del sistema de control.

7.1. Costos Directes

7.1.1. Costos de recursos materials

En la següent taula es mostra els costos de recursos materials que s'han adquirit per aquest treball, especificant la quantitat que es fa servir, el preu unitari i total de cada component que es treballa durant el projecte.

MATERIAL	UNITATS	PREU UNITARI (€/u.)	COST TOTAL (€)
Arduino UNO R3	1	23,70	23,70
Connector USB (2 mascles de tipus A i B)	1	6,99	6,99
Servomotors SunFounder	2	11,99	23,98
Font d'alimentació de 3,3V/5V	1	6,99	6,99
Interruptor (8,5x8,5mm)	1	0,67	0,67
Resistència 330 Ω (pack de 5 u)	5	0,05	0,25
LED vermell	1	0,18	0,18
Cables mascle per protoboard	10	0,293	2,93
Protoboard	1	2,87	2,87
COST TOTAL (€)	-	-	68,56

Taula 7.1. Costos de recursos materials



7.1.2. Costos de recursos humans

En la taula 7.2 s'exposa la dedicació de cada tasca per al desenvolupament d'aquest projecte. Degut a les diferents parts de l'estudi i la seva dificultat, s'ha estimat un preu individual de professional, diferent per a la realització de cada tasca.

TASCA	HORES (h)	COST UNITARI (€/h)	COST TOTAL (€)
Estudi preliminar	15	30	450
Proposta	20	40	800
Disseny definitiu	120	40	4800
Compra material	30	20	600
Proves de la connectivitat dels dispositius	60	30	1800
Programació	100	40	4000
Proves	20	30	600
Memòria	60	35	2100
TOTAL	425 h	-	15.150 €

Taula 7.2. Costos de recursos humans

7.2. Costos indirectes

Es calculen els costos indirectes, estimats com un 10 % dels costos directes. D'aquesta manera es considera els costos relacionats a l'ordinador, smartphone Android, connexió a Internet, llum i altres.



7.3. Cost total del projecte

Finalment, es mostra el preu total del projecte amb l'IVA inclosa.

COST TOTAL (€)	
Costos directes	
Cost de recursos materials	68,56
Cost de recursos humans	15.150 €
Costos indirectes (10%)	1.522 €
TOTAL (AMB IVA)	18.331 €

Taula 7.3. Costos totals del projecte





8. Impacte ambiental

En aquest capítol s'estudia l'impacte ambiental associat al desenvolupament d'aquest projecte, en el que s'han fet servir diferents tipus de components electrònics, i s'avalua les seves conseqüències.

Es treballa amb la placa Arduino, diversos components electrònics, un smartphone i un PC. Per tant, tots aquests generen uns residus electrònics que cal tenir en compte quan s'acaba la vida del producte i prendre unes mesures d'on s'ha de reciclar.

Actualment, els aparells electrònics són cada vegada més abundants i per això provoca un gran augment de residus electrònics. Aquests residus contenen un alt grau d'elements tòxics que poden representar grans riscos a l'ambient i a la salut pública, degut al tipus de materials que estan fets com el plom, arsènic, mercuri, coure, etc.

Per això, s'ha de prendre consciència de la importància de reciclar adequadament cada un dels seus components quan es queden en desús. Degut a que es generen residus que podrien tenir un impacte ambiental. A més a més, la despreocupació de tirar qualsevol dispositiu electrònic a les escombraries sense intentar recuperar les peces útils genera un gran dany al nostre ecosistema.

Afortunadament, la legislació ambiental cada vegada és més exigent i les persones estan més conscienciades del grau d'importància del reciclatge dels components electrònics. Per això existeixen diferents punts de recollida d'aquests materials que s'encarregaran a fer un tractament dels residus. L'objectiu principal d'aquests propòsits és reduir les quantitats de residus i la perillositat dels seus components i regular la gestió per millorar la protecció ambiental, fonamentant la seva reutilització i valoració.





9. Conclusions

En aquest treball final de grau s'ha proposat crear una aplicació de mòbil que pogués controlar una butaca motoritzada, a l'àmbit d'loT, Open Hardware i Open Software. Ha estat una experiència molt enriquidora i valuosa degut a què es desconeixia com procedir a la creació d'una aplicació que permetés el control dels motors des d'un smartphone Android i utilitzant una placa de la família Arduino.

S'han aconseguit tots els objectius generals que es proposaven en aquest projecte. El disseny de l'aplicació ha estat la part més fàcil, ja que es tenia molt clar el que es volia realitzar. No obstant, el desenvolupament i la posada al punt del sistema ha presentat un sèrie de reptes importants.

D'una banda, una de les dificultats més grans d'aquest projecte ha sigut aconseguir la connectivitat d'Internet amb els dispositius proposats. Inicialment, es va comprar la placa Arduino UNO WIFI, però no era compatible per establir la connexió amb l'aplicació Blynk. Per aquest motiu, es va descartar aquesta opció i es va optar per comprar la placa Arduino UNO R3 i el mòdul de WIFI (ESP8266).

Després de realitzar les connexions i configuracions adequades, es va aconseguir que el mòdul ESP8266 obtingués una IP d'una WiFi, gràcies a la seva configuració amb una connexió USB-sèrie. Però quan es feia un RESET, el mòdul s'havia de reconfigurar cada vegada, degut a què no s'havien guardat totes les comandes prèvies. Per aquesta raó, no va ser possible assolir la connectivitat a Internet de la placa Arduino. Per continuar amb aquesta aproximació, es requeria molts coneixements addicionals que no es podien assolir en un termini tant curt com el que es disposava en aquest projecte.

Finalment, després dels dos intents anteriors, es va decidir utilitzar la placa Arduino UNO R3, connectada a través del cable USB al PC, per intentar solucionar aquest problema. Després d'un estudi exhaustiu de les llibreries de comunicació, es va trobar una que permetia aquest accés a Internet de la placa Arduino a través del PC.

Un cop que es va seleccionar aquesta llibreria, es va trobar unes complicacions a causa de què la última versió de la llibreria Blynk era incompleta. El principal problema va ser que es va descarregar la última versió de la llibreria Blynk 4.2, i es va observar que no es podia connectar correctament. Per això, es va fer una revisió profunda on es va detectar que faltava un fitxer "*com2tcp*". Aleshores, com que es disposava d'una versió anterior, 4.0, es va poder instal·lar i aconseguir que la connexió funcionés adequadament. Aquest tipus de problemes a vegades apareixen al treballar a l'àmbit d'loT i d'Open Source.



D'altra banda, en el desenvolupament del codi de l'aplicació han sorgit una sèrie de reptes. El fet de treballar amb la plataforma Blynk i la placa d'Arduino ha requerit ajustar i sincronitzar el codi, de manera que realitzés tot el que es proposava en aquest projecte. Tot i que els widgets són fàcils d'incorporar a l'aplicació, requereixen una configuració molt particular que complica la codificació del programa.

Per últim, el control dels dos servomotors no ha estat una tasca senzilla, ja que els codis estan pocs estandarditzats. Primerament es va optar per utilitzar la llibreria que disposa l'Arduino IDE, però no era suficient degut a què és una llibreria molt limitada en quant els paràmetres de control. Per tant, es va procedir a buscar altres alternatives. Gràcies a què es treballa en l'àmbit d'IoT i d'Open Source, ha estat possible obtenir una llibreria que permet l'ajust dels paràmetres de controls que es necessiten per a aquest projecte.

El resultat d'aquesta part del treball ha estat el disseny i desenvolupament de l'aplicació MA_BUTACA, que permet controlar i monitoritzar les posicions angulars dels servomotors de la butaca, i també les diverses opcions que ofereix aquesta aplicació.

Es vol donar èmfasi al nom de l'aplicació MA_BUTACA ("Mobile-Aduino BUTACA") pel fet que ha sigut tot un desafiament aconseguir la connexió entre aquests dos dispositius, la seva configuració i el seu paper per poder controlar els servomotors. També es vol destacar en aquest nom, que és una abreviatura de la "meva butaca", ja que aquesta aplicació està dirigida per a tot tipus d'usuari que vulguin seleccionar la seva butaca a la posició desitjada i de confort.

Com a part final del projecte, s'ha desenvolupat un prototipus demostrador per poder visualitzar les diferents opcions de control que ofereix MA_BUTACA. Es pot afirmar que s'han acomplert tots els objectius específics que es proposaven.

Finalment, amb l'experiència d'utilitzar el mòdul ESP8266, la possibilitat de disposar d'un servidor local de Blynk i altres opcions de millora, es plantegen nous reptes i noves portes per a un possible estudi futur. Per això, en el següent capítol "Extensions futures", s'exposa una sèrie de propostes per millorar el sistema desenvolupat en aquest projecte.



10. Extensions futures

S'ha creat una aplicació de mòbil que permet controlar de manera individualitzada els motors d'una butaca. Els components més importants que s'han emprat al llarg del projecte són la placa Arduino, els dos servomotors i el desenvolupador Blynk. Durant el seu desenvolupament s'ha aconseguit les funcionalitats que s'han plantejat, però s'han identificat unes opcions de millora del sistema proposat en aquest projecte i que es podrien realitzar en un futur.

A continuació, es comenten diferents aspectes d'estudi a millorar:

- Crear un servidor local del Blynk.
- Disposar del mòdul ESP8266 com a punt d'accés WiFi per la placa Arduino UNO R3.
- Crear un identificador de la butaca perquè l'usuari pugui seleccionar la butaca que vol controlar.
- La usabilitat per a tot tipus d'usuaris.

10.1.Servidor local: Blynk server

L'aplicació dissenyada disposa d'un Cloud Server propi de Blynk que permet establir la comunicació entre la placa Arduino i el desenvolupador Blynk. El fet d'emprar el Cloud Server no seria una bona opció per si algun dia es vol comercialitzar la butaca motoritzada amb la possibilitat de ser controlada a través d'una aplicació des d'un smartphone.

Per això es proposa l'opció d'emprar un Blynk Server Local, que és un servidor basat en Java Netty Open Source, i és responsable de reenviar missatges entre aplicacions de mòbil de Blynk i diferents microcontroladors com l'Arduino.

Les raons principals per utilitzar el servidor local de Blynk són:

- Millor seguretat: el creador de l'aplicació té el coneixement sobre el servidor, permet configurar les polítiques de seguretat segons les seves necessitats específiques. També disposa l'opció de fer accessible només dins de la xarxa privada.
- Millor estabilitat: No hi ha la necessitat de dependre del núvol, disposa del control absolut del servidor.



- Màxima privacitat: Totes les dades s'emmagatzemen localment i no és comparteixen amb ningú.

Tot i que és una extensió futura, s'ha estudiat el procediment per obtenir el servidor local de Blynk.

Per configurar-lo, es requereix instal·lar el software Java 8 (OpenJDK, Oracle) i descarregar l'última versió d'arxius del servidor local de Blynk. Seguidament, s'obre la finestra de l'administrador on s'ha d'escriure una sèrie de comandes:

1. Es comprova que s'ha instal·lat el Java 8: *java -version*
2. Es col·loca el *path* on està l'arxiu ".jar": *cd Users/User/Blynk/Server*
3. S'introdueix la comanda per executar el servidor local de Blynk:

java -jar server-0.21.4.jar -dataFolder /path

Per defecte del servidor es fan servir aquests ports:

- Hardware: 8442
- Blynk App: 8443

Un cop configurat, es procedeix a habilitar la comunicació entre el desenvolupador Blynk amb el servidor local i es necessita especificar a l'aplicació quin servidor es connectarà. Aquest servidor es carrega amb l'adreça IP i el port 8443.

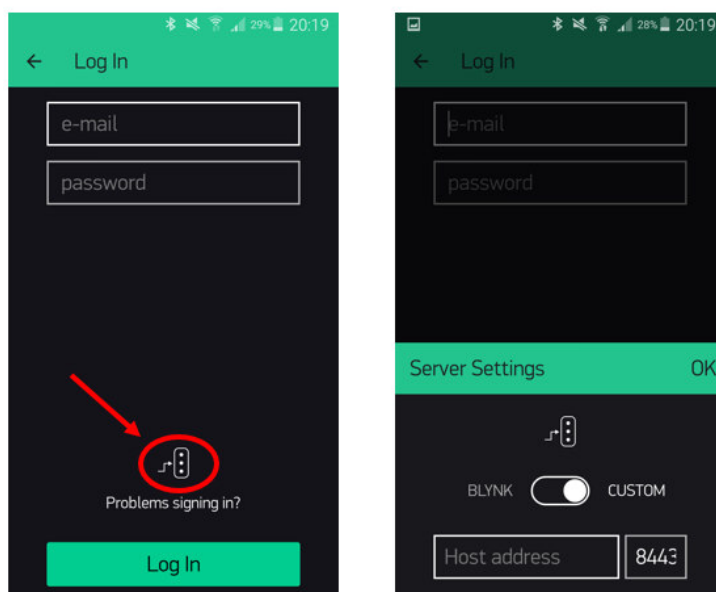


Figura 10.1. Configuració servidor local Blynk



10.2.Mòdul ESP8266

El desenvolupament final d'aquest projecte ha considerat la placa Arduino connectada al PC en tot moment, de manera que no ha sigut necessari disposar d'un mòdul WiFi per poder aconseguir la comunicació a través d'Internet entre l'Arduino i el desenvolupador Blynk.

En un futur es valora incorporar aquest mòdul ESP8266 (Fig. 10.2) com a punt d'accés WiFi, per prescindir del PC connectat a la placa Arduino.

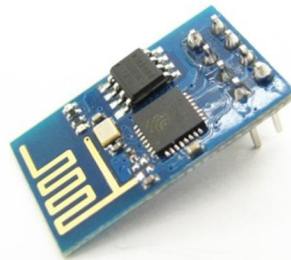


Figura 10.2. Mòdul ESP8266

En el subapartat 3.3.2 s'explica que es va poder configurar aquest mòdul a través d'una sèrie de comandes AT. No obstant, va faltar completar la combinació entre la placa Arduino i el mòdul ESP8266. Per això, es proposa aquesta opció a millorar en un futur.

10.3. Identificador de butaca

L'aplicació MA_BUTACA permet el control d'una placa Arduino específica, que s'indica a través d'un AuthToken que s'incorpora en el codi del programa a l'Arduino IDE i de l'aplicació Blynk. Aquest control està orientat per governar una butaca determinada i no permet seleccionar entre vàries.

Per això, en un futur es proposa introduir un identificador a la placa de control de la butaca. D'aquesta manera, l'smartphone disposarà d'un widget que permetrà introduir aquest identificador, que es comunicarà amb el sistema de control. El sistema llegirà l'identificador que envia l'smartphone, i si aquest coincideix amb el que té assignat la butaca, es procedirà en fer l'ús de l'aplicació. En cas contrari, no farà res.



10.4. Usabilitat per a tot tipus d'usuaris

L'aplicació que s'ha desenvolupat permet ser manipulada per diferents tipus d'usuaris. No obstant no s'ha tingut en compte per als usuaris amb deficiència visual. Per tant, no es pot afirmar que tingui una usabilitat per a qualsevol tipus d'usuari.

Aquest fet motiva a que es proposi un estudi futur, de manera que s'adapti a les necessitats de tot tipus d'usuaris i sigui accessible.

En el cas dels usuaris amb deficiència visual es proposa d'emprar els mateixos widgets que els de l'aplicació MA_BUTACA, però es configuraria d'una manera diferent. És a dir, incorporant un reconeixedor de veu, ja que es tractaria que l'usuari controli per veu els widgets del sistema. També l'aplicació informaria per veu a l'usuari de la realització de cada acció.

Per exemple, l'usuari diria "mou el slider de reposapeus a 40°" i seguidament l'aplicació ho interpretaria aquesta ordre i actuaria el motor.

També se li proporcionarà el manual disponible en veu o escrit en Braille, explicant com funciona aquest sistema amb el reconeixement de veu. Així tindrien la informació per fer l'ús de l'aplicació, que fa cada widget i quin tipus de so i/o missatge de veu emetrà al ser pulsat.



Agraïments

Vull donar les gràcies al professor Emili Lupon, del departament d'Enginyeria Electrònica, per la dedicació, guiar-me i aconsellar-me al llarg del projecte.

També vull agrair a la meva família, sobre tot als meus pares i a l'Adrià, i als amics per tot el seu suport, paciència i comprensió durant tots els meus estudis i especialment en el treball final de grau.





Bibliografia

Referències bibliogràfiques

- [1] Baiborodin, P., & Dumanskiy, D. (sense data). *Blynk*. Recollit de Blynk: <http://www.blynk.cc/>
- [2] Banzi, M., & Kushner, D. (sense data). *Arduino*. Recollit de Arduino: <https://www.arduino.cc/>
- [3] *BLYNK-About us*. (sense data). Recollit de <http://www.blynk.io/about>
- [4] *Figueras International Seating*. (sense data). Recollit de Figueras International Seating: <http://www.figueras.com/es/>
- [5] Friedman, M., & Abelson, H. (sense data). *APP INVENTOR*. Recollit de APP INVENTOR: <http://appinventor.mit.edu/explore/>
- [6] García. (18 / Maig / 2016). *Tutorial para configurar el modulo WIFI ESP8266 con Arduino UNO y enviar datos por la WEB (Comandos AT)*. Recollit de <https://www.youtube.com/watch?v=7qXcTBHLCRc>
- [7] García, A. (sense data). *DIYmakers*. Recollit de DIYmakers: <http://diymakers.es/>
- [8] Gu, P. (2014). *NURDs*. Recollit de NURDs: <https://nurdspace.nl/ESP8266>
- [9] *KICK STARTER*. (sense data). Recollit de https://www.kickstarter.com/projects/167134865/blynk-build-an-app-for-your-arduino-project-in-5-m/creator_bio
- [10] MAKER.IO Staff. (2 / Febrer / 2016). *MAKER.IO*. Recollit de <https://www.maker.io/en/interviews/2016/interview-with-pavel-baiborodin---founder-of-blynk>
- [11] *MollyFlex*. (sense data). Recollit de MollyFlex: <http://www.mollyflex.es/accesorios/542/linea-butacas-relax>
- [12] SunFounder. (24 / Juliol / 2015). *SunFounder Kit Tutorial for Arduino - Servo*. Recollit de SunFounder Kit Tutorial for Arduino - Servo: <https://www.youtube.com/watch?v=t1hmZlZrM8>



- [13] Upton, E. (sense data). *RASPBERRY PI*. Recollit de RASPBERRY PI:
<https://www.raspberrypi.org/>

